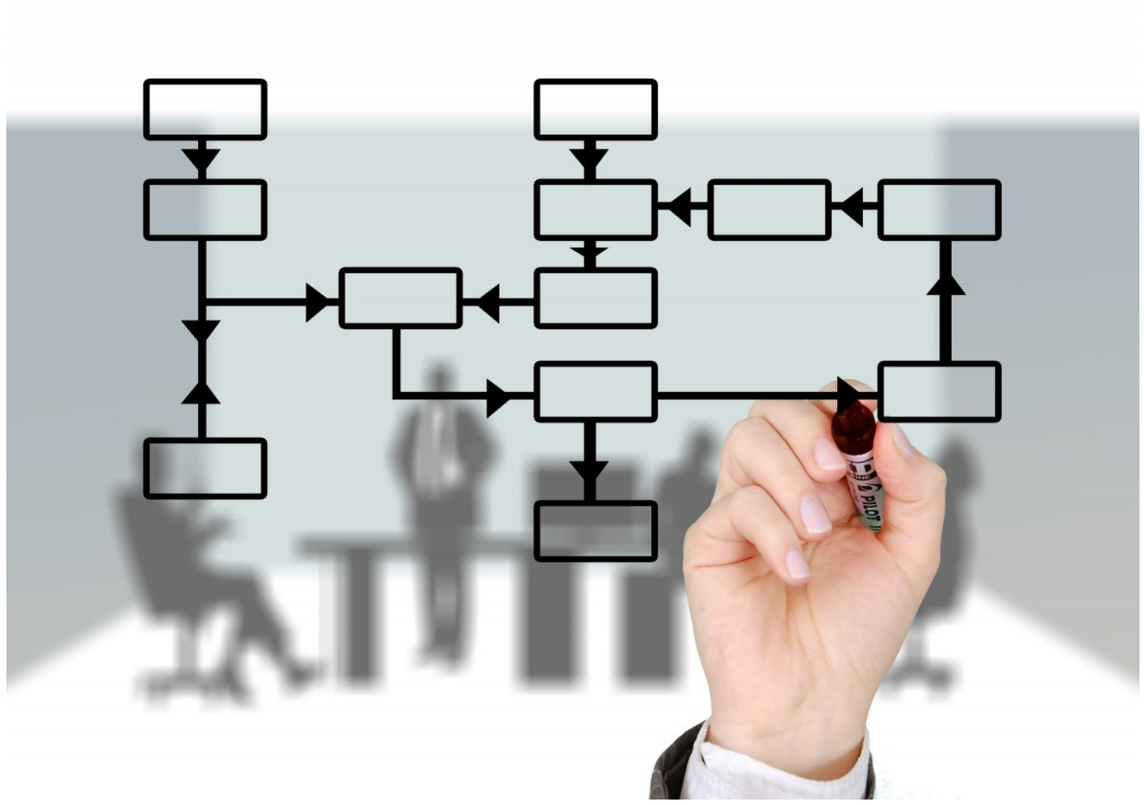
BUSINESS PROCESS ANALYTICS IN R

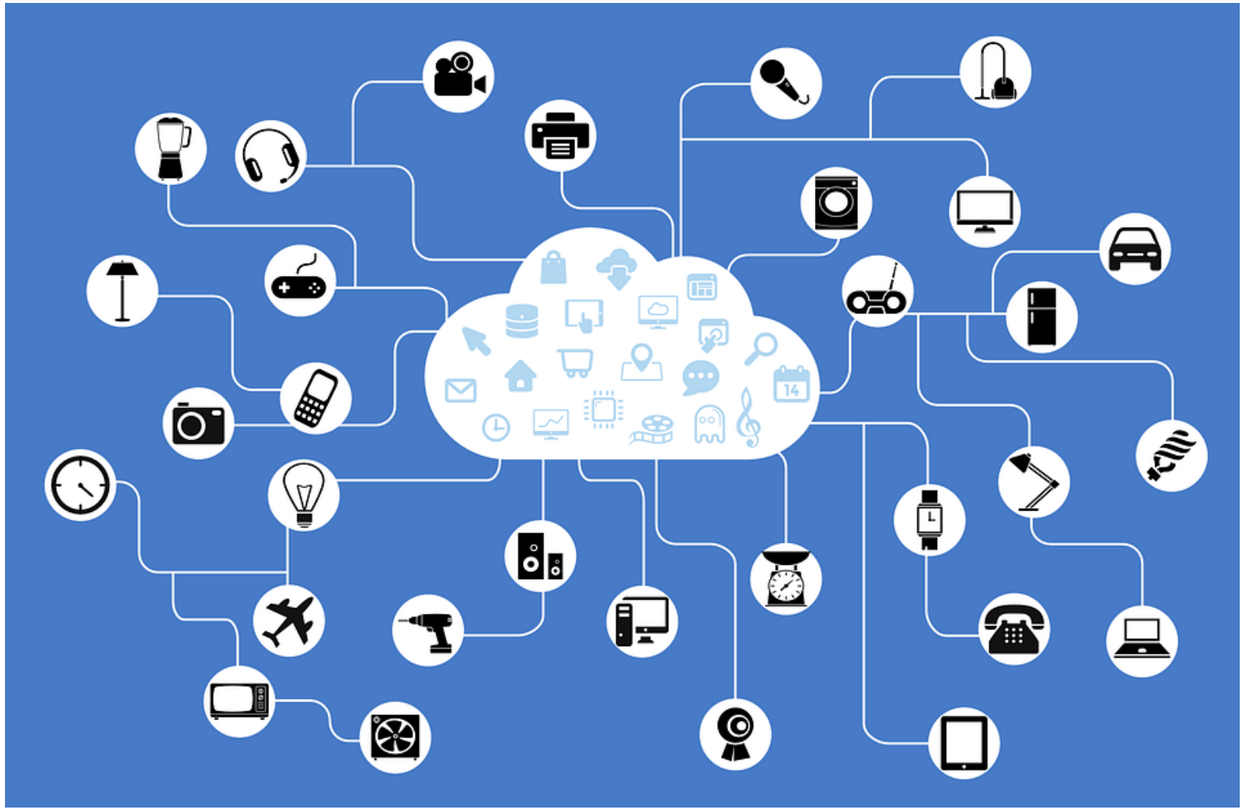
Introduction to Process Analytics

Gert Janssenswillen
Creator of bupaR

Business Processes



Event data

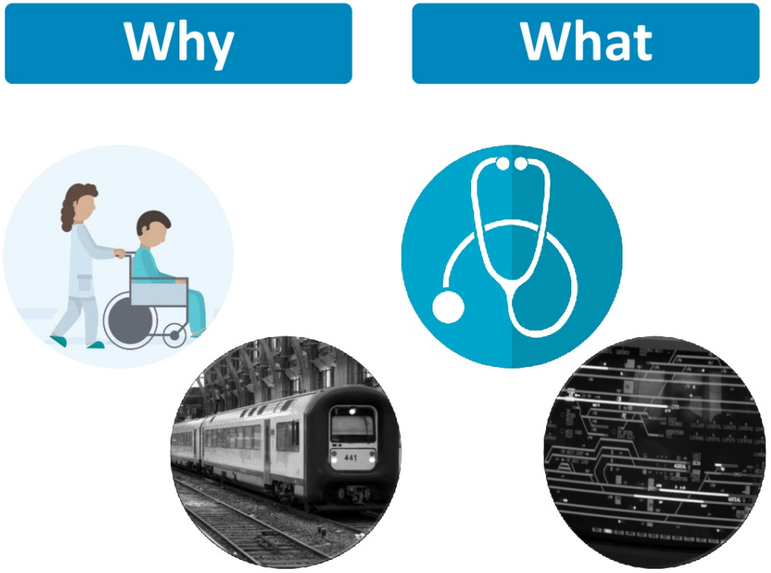


Process data

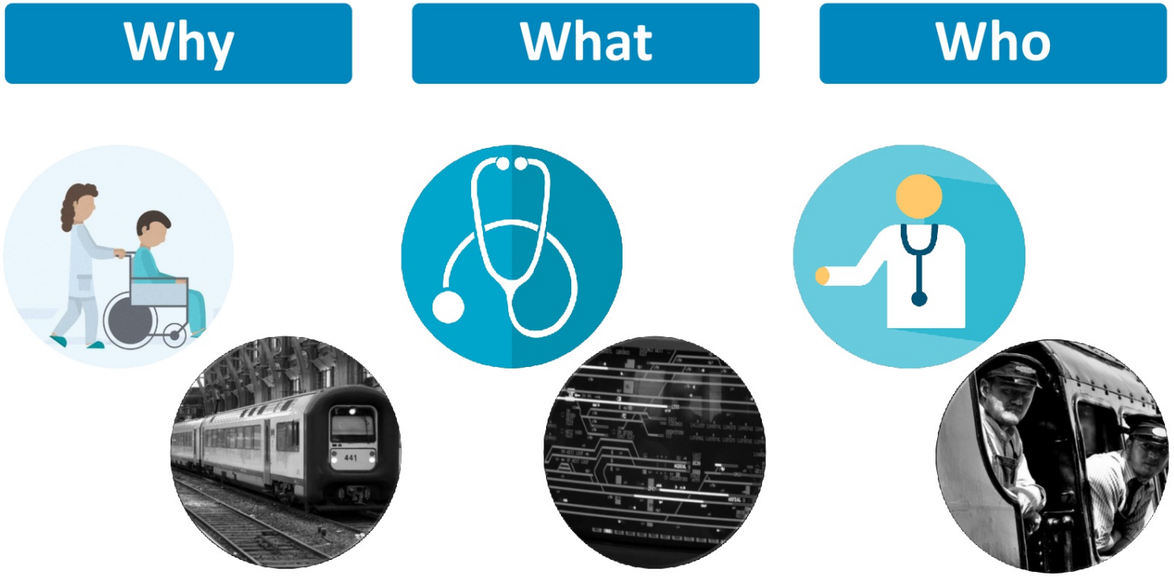
Why?

Why **What** **Who**

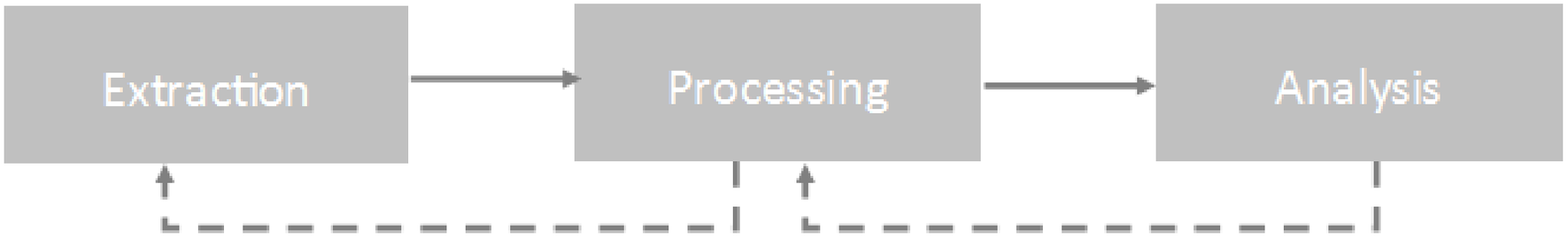
What?



Who?



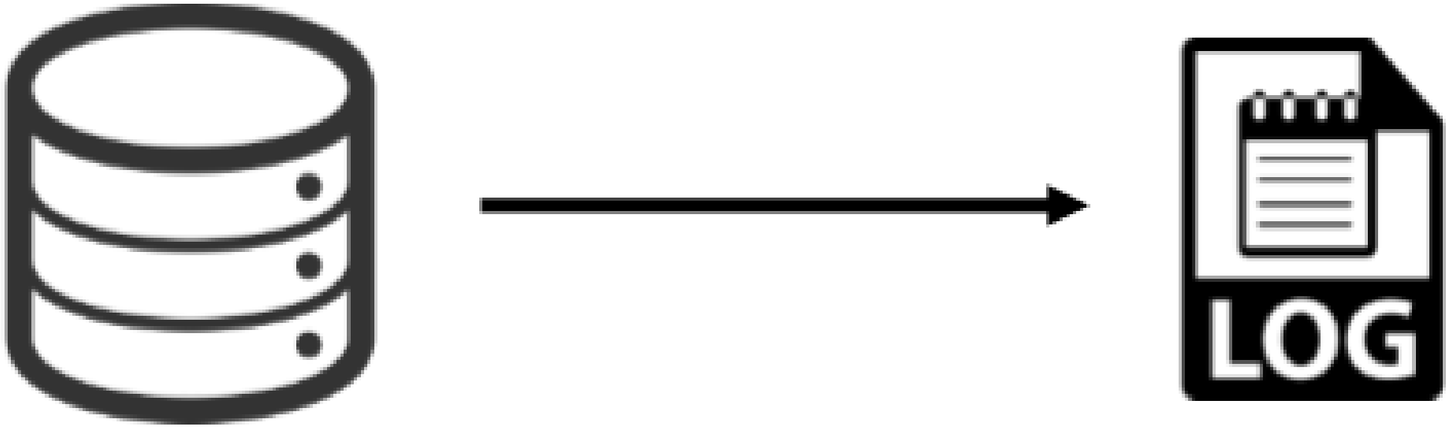
Process analysis workflow



1. **Extraction**: transform raw data into event data
2. **Processing**: enrich and filter event data
3. **Analysis**: gain useful insights in the process

Event Data Extraction

From raw data to event data



Event Data Preprocessing

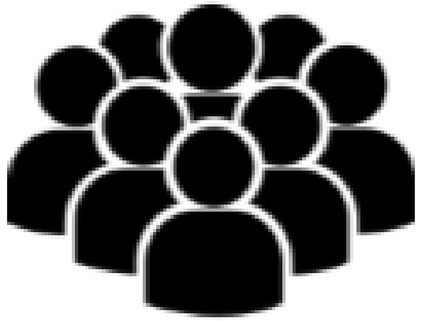
Aggregation: remove redundant details

Enrichment: add useful data attributes

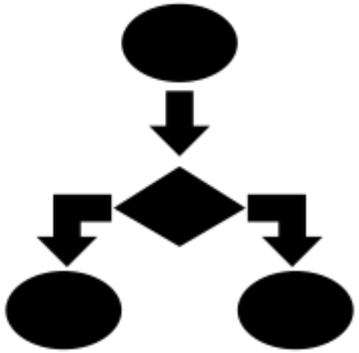
Filtering: focus your analysis

Event Data Analysis

Organizational



Control-flow



Performance

And also

- Multivariate analysis
- Include additional data attributes

BUSINESS PROCESS ANALYTICS IN R

Let's practice!

BUSINESS PROCESS ANALYTICS IN R

Activities as cornerstones of processes

Gert Janssenswillen
Creator of bupaR

Example: Online learning

A first glimpse of the event log

Getting an idea about the event log scope

- How many **cases** are described?
- How many distinct **activities** are performed?
- How many **events** are recorded?
- What is the **time period** in which the data is recorded?

A first glimpse of the event log

```
library(bupaR)
```

This information can be viewed by printing the summary of an event log

```
summary(learning)
```

or using count functions.

```
> n_cases(learning)
498
> n_activities(learning)
10
> n_events(learning)
3645
```

Activities

Activities describe the flow of the process

- Which actions are performed?
- In what order are they performed?

Exploring activities

```
> activity_labels(learning)
```

```
 [1]  "Consult Dictionary"   "Consult Theory Pages" "Exercise 1"
 [4]  "Exercise 2"           "Exercise 3"           "Exercise 4"
 [7]  "Exercise 5"           "Exercise 6"           "Exercise 7"
 [10] "Assessment"
```

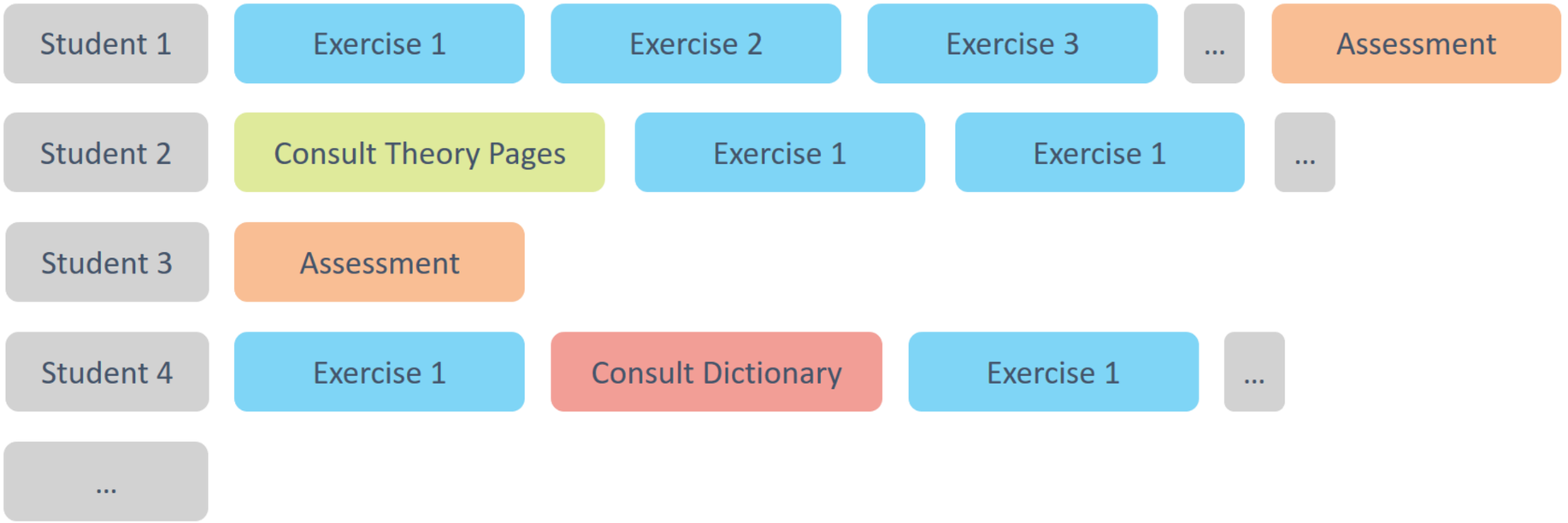
Exploring activities

```
activities(learning)
```

```
# A tibble: 10 x 3
   action               absolute_frequency relative_frequency
    <chr>                             <dbl>              <dbl>
 1 Exercise 1                          516              0.142
 2 Assessment                          498              0.137
 3 Exercise 2                          493              0.135
 4 Exercise 4                          442              0.121
 5 Exercise 3                          436              0.120
 6 Exercise 5                          360              0.0988
 7 Exercise 6                          302              0.0829
 8 Exercise 7                          299              0.0820
 9 Consult Dictionary                  165              0.0453
10 Consult Theory Pages                134              0.0368
```

Exploring sequences of activities

Each case is described by a sequence of activities, its **trace**.



Exploring sequences of activities

- A frequency table of traces can be retrieved with the `traces` function

```
traces(learning)
```

- They can be visualized using the `trace_explorer` function

```
trace_explorer(learning)
```

BUSINESS PROCESS ANALYTICS IN R

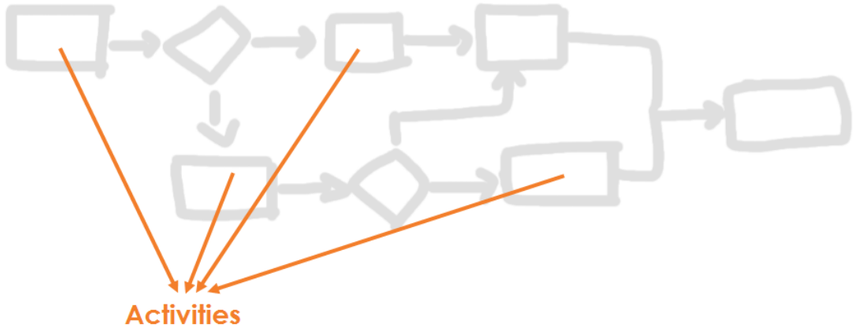
Let's practice!

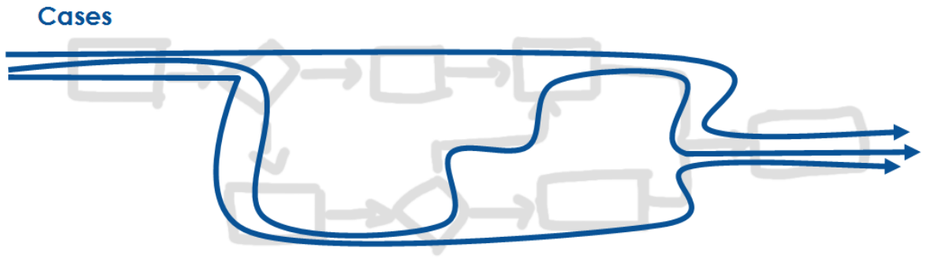
BUSINESS PROCESS ANALYTICS IN R

Components of process data

Gert Janssenswillen
Creator of bupaR

Cases and activities



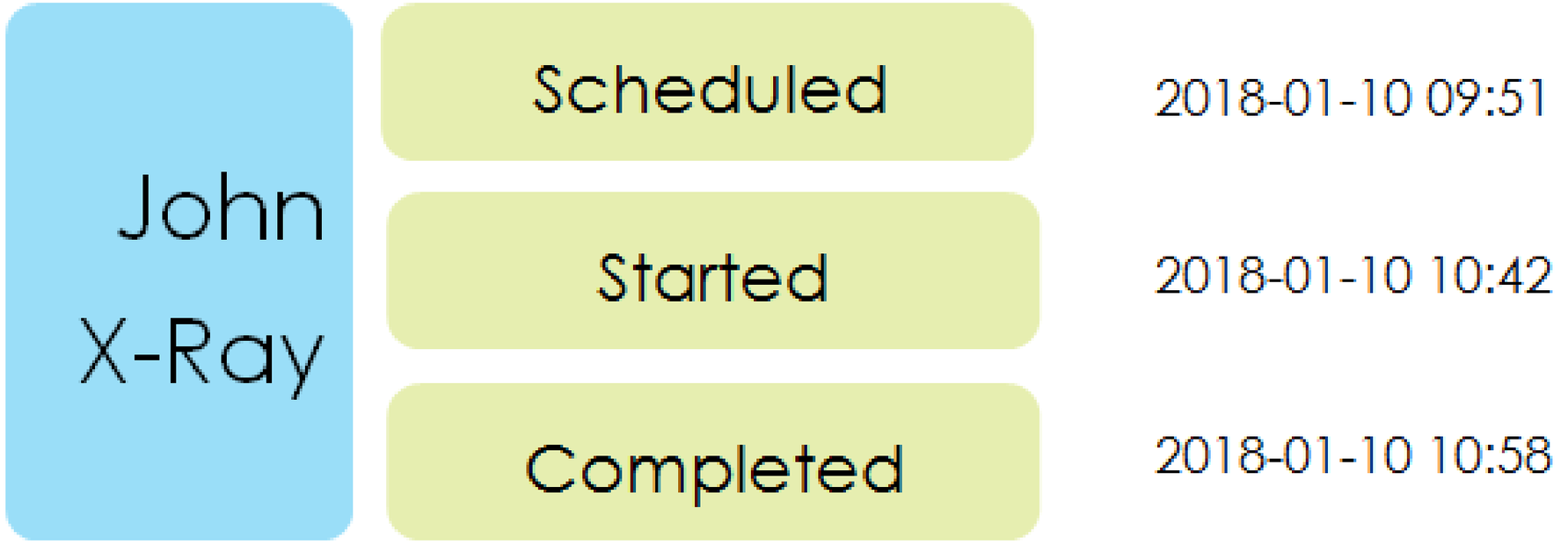


Activity instances

Activity instance = **occurence** of an activity

	Patient	Activity	Started at
1	John	Registration	2018-01-10 09:41
2	Emily	Registration	2018-01-10 10:36
3	John	X-Ray	2018-01-10 10:42

Events


John
X-Ray
Scheduled
Started
Completed
2018-01-10 09:51
2018-01-10 10:42
2018-01-10 10:58

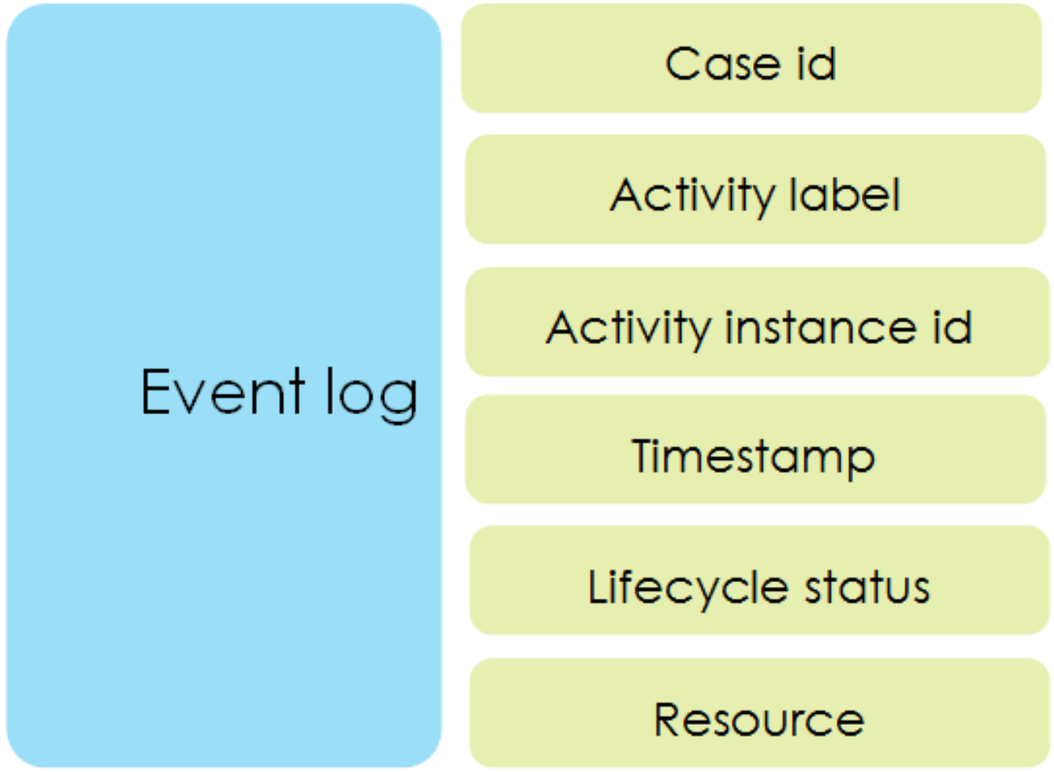
Event log

Instance	Patient	Activity	Status	Time
1	John	Registration	Start	2018-01-10 09:41
3	John	X-Ray	Schedule	2018-01-10 09:51
2	Emily	Registration	Start	2018-01-10 10:36
3	John	X-Ray	Start	2018-01-10 10:42
3	John	X-Ray	Complete	2018-01-10 10:58

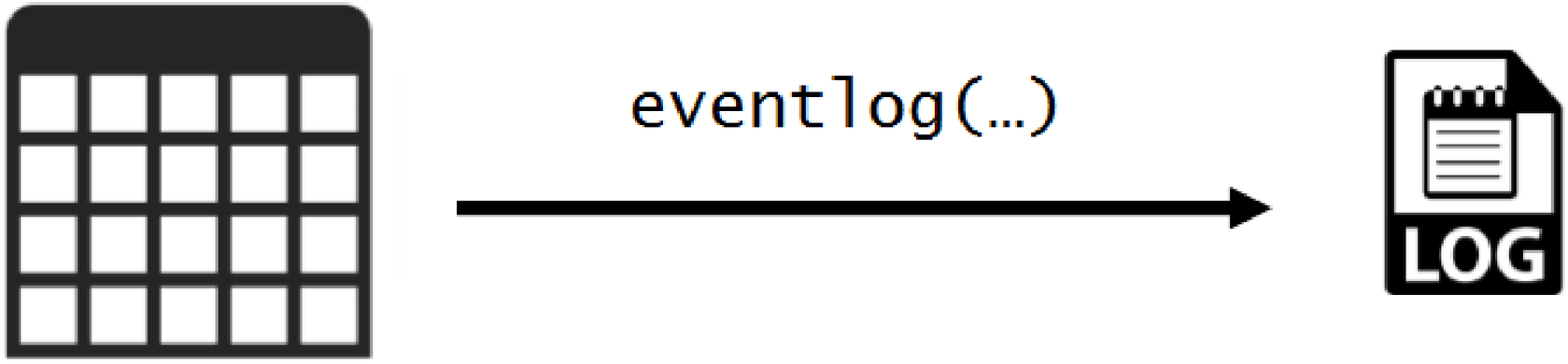
Resources

Instance	Patient	Activity	Status	Time	Resource
1	John	Registration	Start	2018-01-10 09:41	Mr. Owens
3	John	X-Ray	Schedule	2018-01-10 09:51	Dr. Russell
2	Emily	Registration	Start	2018-01-10 10:36	Mr. Fleming
3	John	X-Ray	Start	2018-01-10 10:42	Dr. Russell
3	John	X-Ray	Complete	2018-01-10 10:58	Dr. Russell

Recap: event log



Create event log object




```
event_data %>%
    eventlog(case_id = "patient",
             activity_id = "handling",
             activity_instance_id = "handling_id",
             timestamp = "time",
             lifecycle_id = "registration_type",
             resource = "employee")
```

BUSINESS PROCESS ANALYTICS IN R

Let's practice!

BUSINESS PROCESS ANALYTICS IN R

Organizational analysis

Gert Janssenswillen
Creator of bupaR

Looking at the actors in the process

- Who executes the work?
- Who specializes in certain task?
- Is there a risk of brain drain?
- Who transfers work to whom?

Data: Hospital process

Who executes the work?

Resources labels

```
resource_labels(log_hospital)

 [1] Clerk Susan
 [2] Dr. Sandra
 [3] Dr. Lindsey
 [4] Dr. John
 [5] Nurse Carol
 [6] Clerk Kimberly
 [7] Nurse William
 [8] Nurse James
 [9] Emergency Dr. Helen
[10] Emergency Nurse Laura
[11] Emergency Nurse Robert
[12] Emergency Nurse David
```

Who executes the work?

Resource frequencies

```
resources(log_hospital)
# A tibble: 12 x 3
   employee               absolute_frequency relative_frequency
   <fct>                               <int>              <dbl>
 1 Dr. John                             1101             0.189
 2 Dr. Lindsey                          1055             0.181
 3 Dr. Sandra                            955             0.164
 4 Clerck Kimberly                       694             0.119
 5 Clerck Susan                          677             0.116
 6 Nurse William                         345             0.0591
 7 Nurse Carol                           313             0.0536
 8 Nurse James                           263             0.0451
 9 Emergency Dr. Helen                   210             0.0360
10 Emergency Nurse Laura                 145             0.0249
11 Emergency Nurse Robert                 68             0.0117
12 Emergency Nurse David                   9             0.00154
```

Resource-activity Matrix

Specialization and brain drain

Specialization

- When a person only performs a **single** activity

Specialization and brain drain

Specialization

- When a person only performs a **limited set of activities**

Brain drain

- When an activity is performed by only a **limited set of resources**

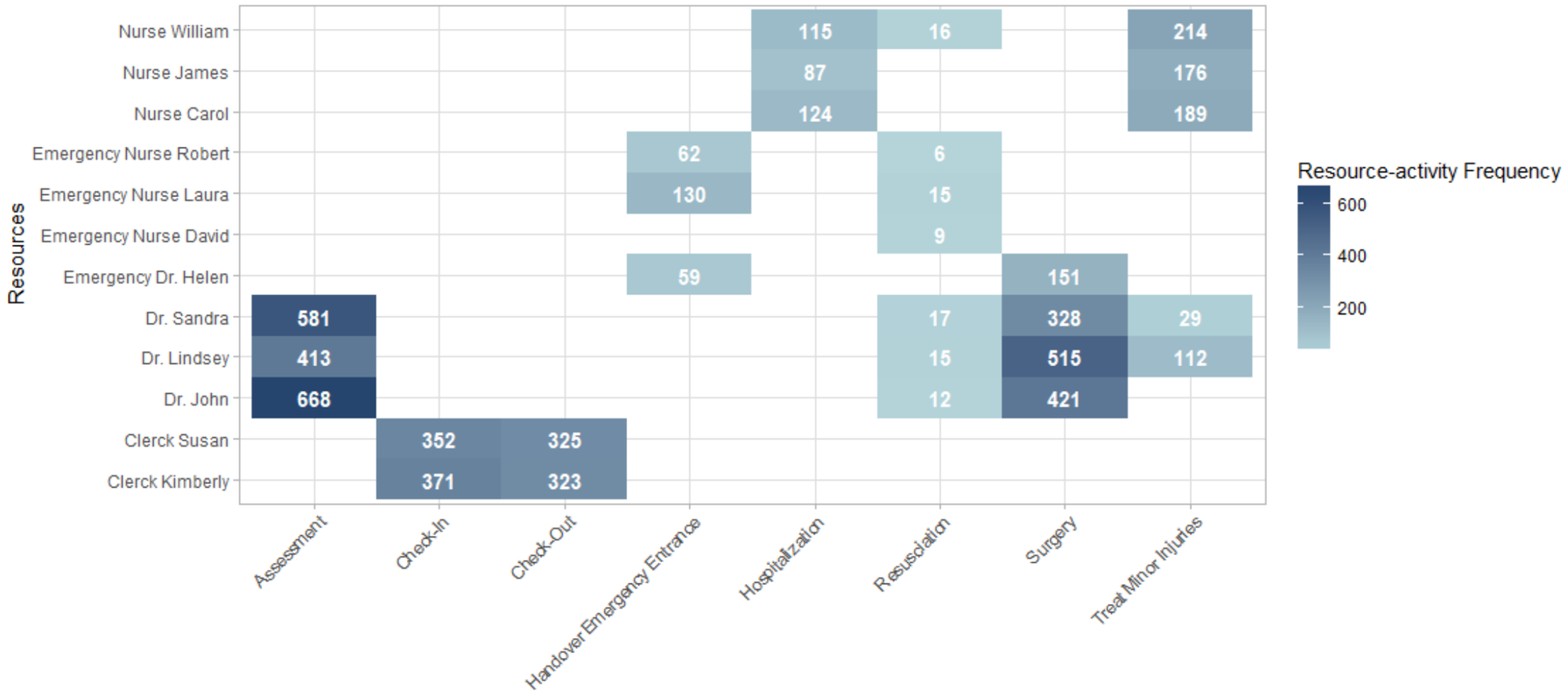
Resource activity matrix

```
log_hospital %>%
    resource_frequency(level = "resource-activity")
```

```
log_hospital %>%
    resource_frequency(level = "resource-activity") %>% plot()
```

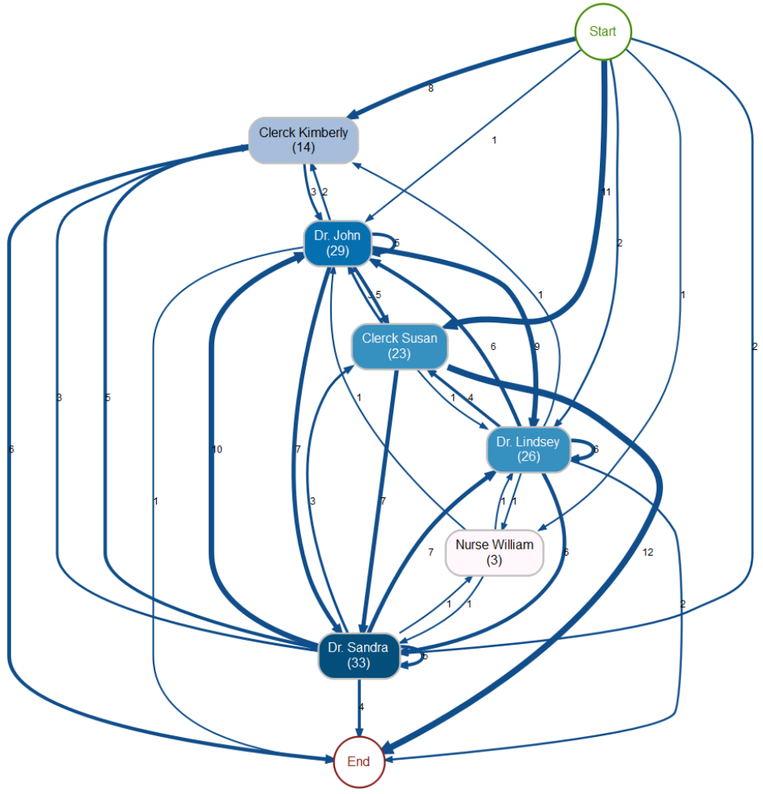
`resource_frequency` is a *process metric*, where the `level` argument indicates at which level of detail you want to calculate it.

Resource activity matrix: example



Who transfers work to whom?

```
resource_map(log_hospital)
```



BUSINESS PROCESS ANALYTICS IN R

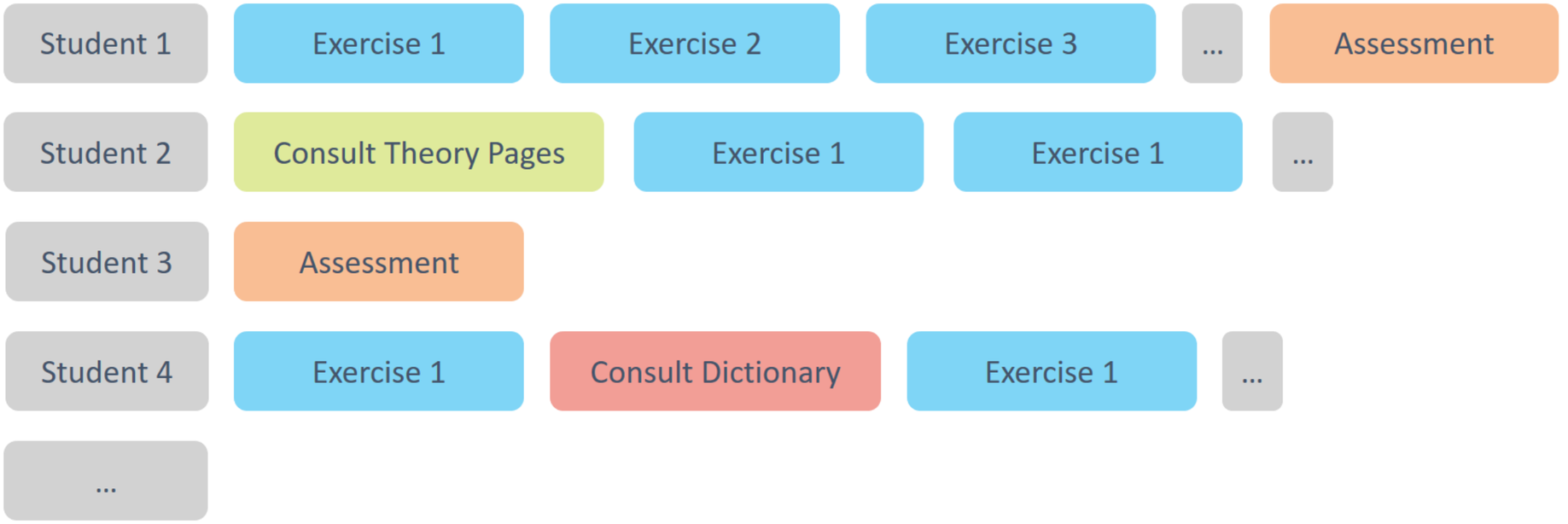
Let's practice!

BUSINESS PROCESS ANALYTICS IN R

Structuredness

Gert Janssenswillen
Creator of bupaR

Control-flow



Further analysis

Metrics

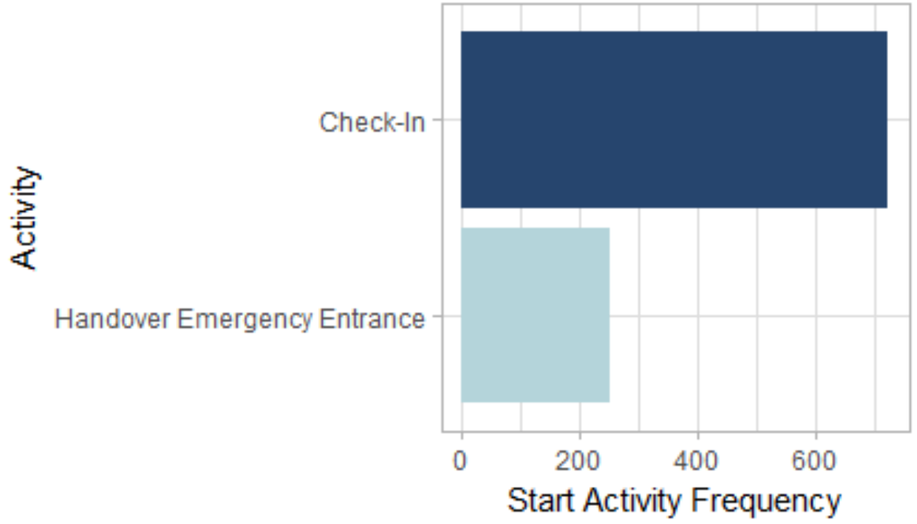
- Entry and exit points
- Length of cases
- Presence of activities
- Rework

Visuals

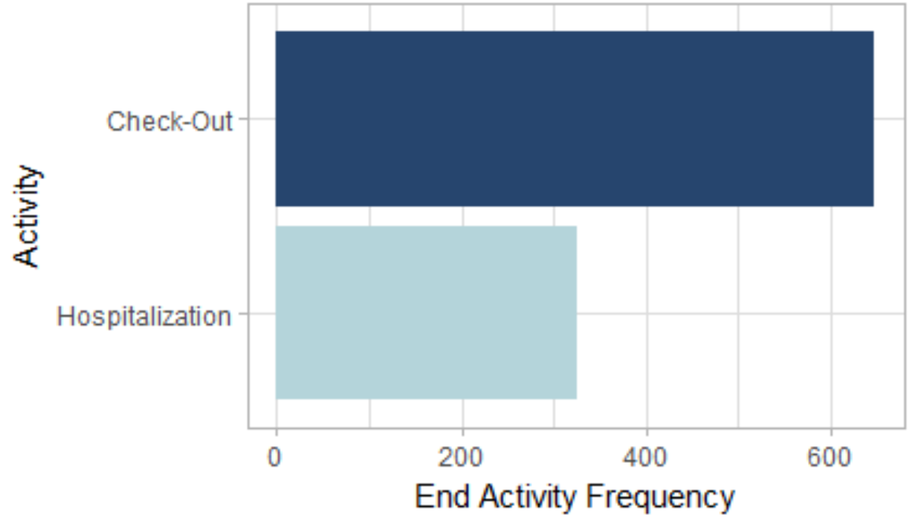
- Process map
- Trace explorer
- Precedence matrix

Entry & Exit points

```
log_healthcare %>%
    start_activities("activity") %>%
    plot()
```

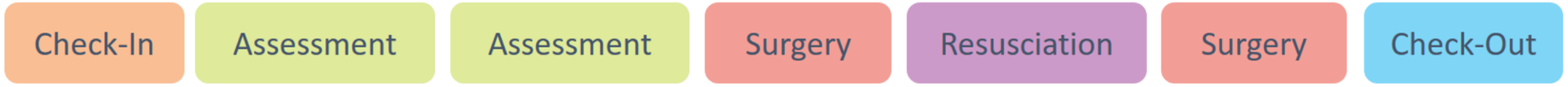



```
log_healthcare %>%
    end_activities("activity") %>%
    plot()
```



Rework

An example patient history



- Repetitions
  - Surgery > ... > Surgery
- Self-loop
  - Assessment > Assessment

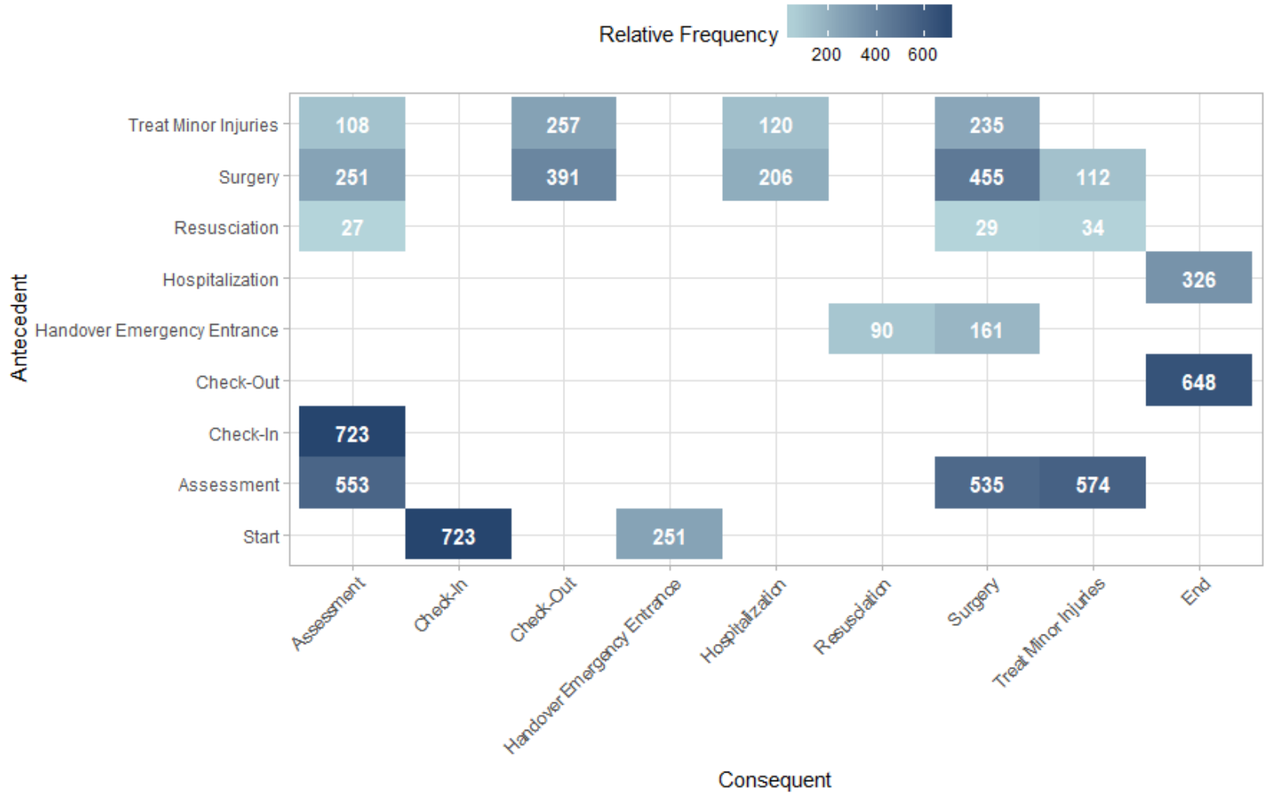
Precedence matrix

Precedence matrix

Creating precedence matrices

```
eventlog %>% precedence_matrix(type = "absolute") %>% plot()
```

Precedence matrix Example



BUSINESS PROCESS ANALYTICS IN R

Let's practice!

BUSINESS PROCESS ANALYTICS IN R

Performance analysis

Gert Janssenswillen
Creator of bupaR

Performance analysis

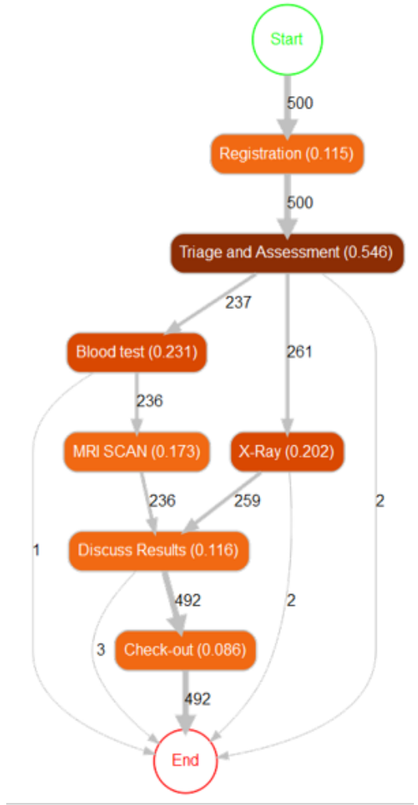
Visuals

- Performance process map
- Dotted chart

Metrics

- Throughput time
- Processing time
- Idle time

Performance process map



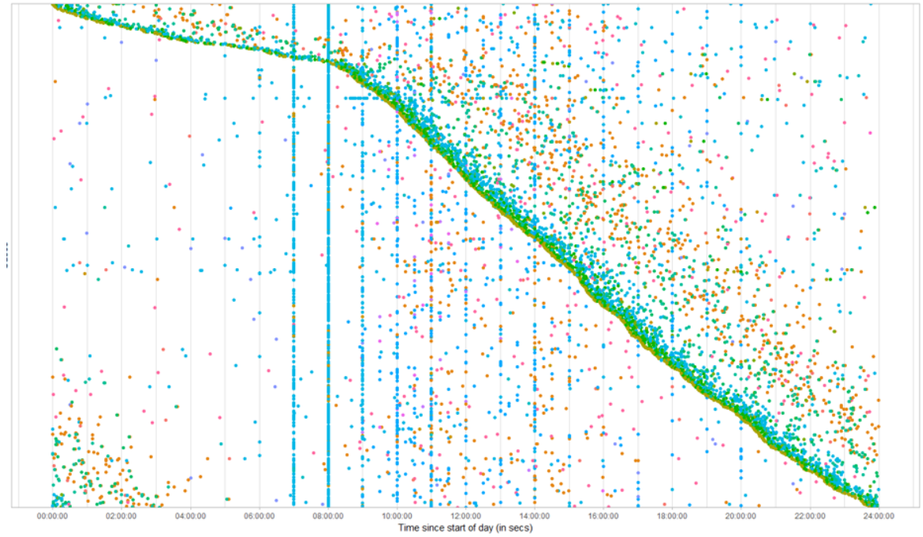
A *normal* process map

```
eventlog %>%
    process_map(type = frequency())
```

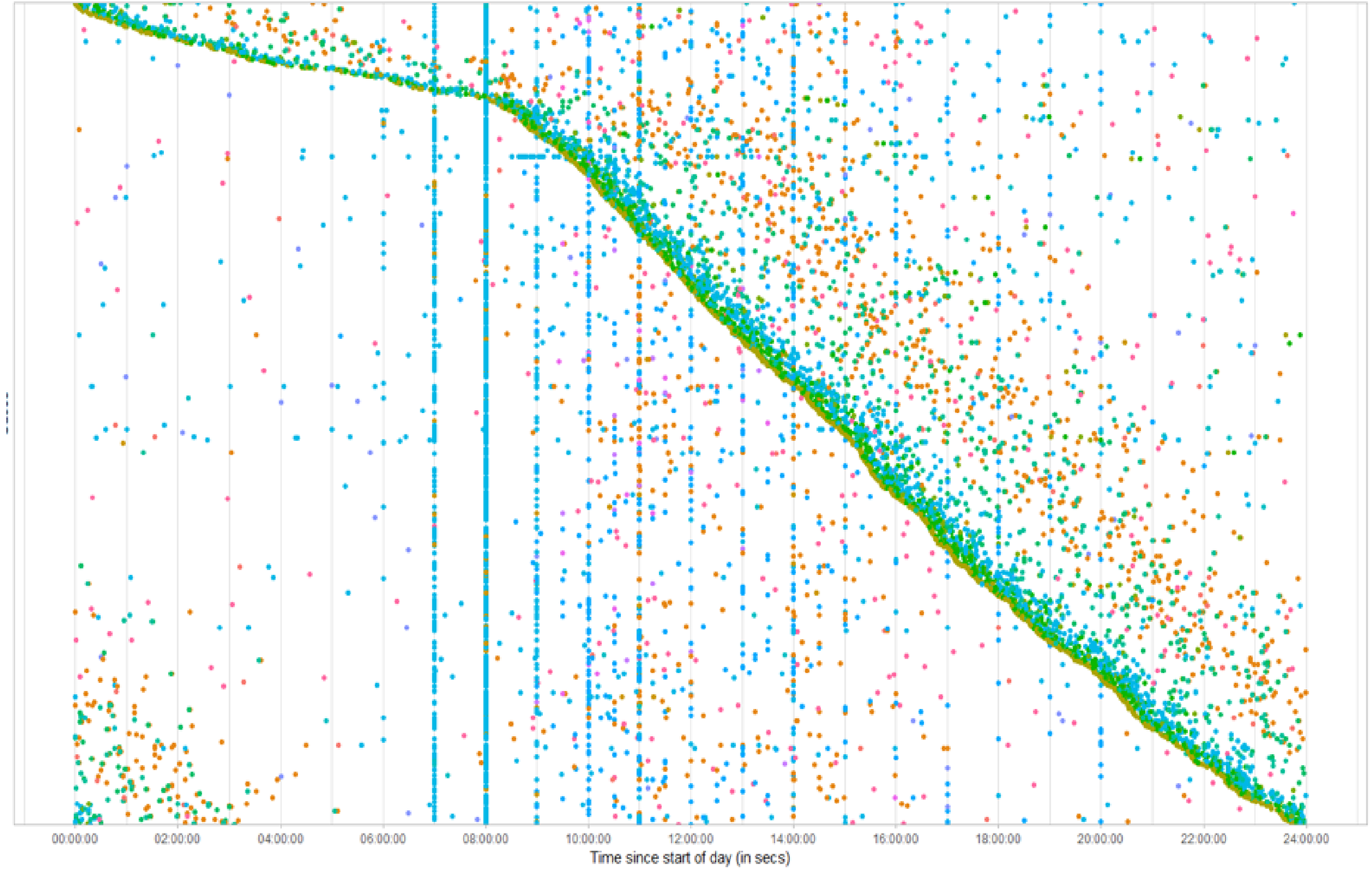
A *performance* process map

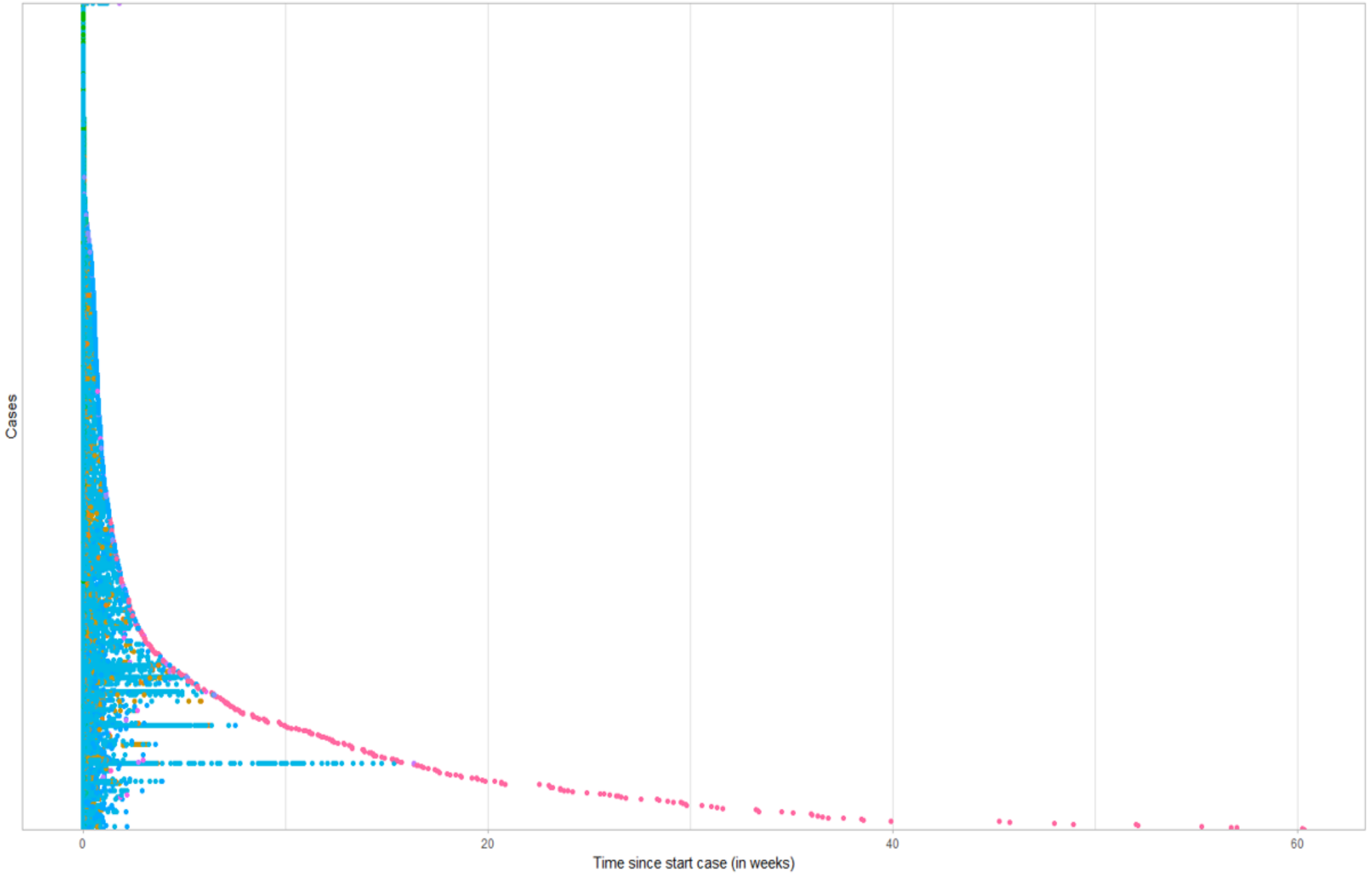
```
eventlog %>%
    process_map(type = performance())
```

Dotted chart



- each dot represents activity
- x-axis: time
- y-axis: cases


00:00:00
02:00:00
04:00:00
06:00:00
08:00:00
10:00:00
12:00:00
14:00:00
16:00:00
18:00:00
20:00:00
22:00:00
24:00:00
Time since start of day (in secs)


Cases
0
20
40
60
Time since start case (in weeks)

Performance metrics

- `throughput_time`
- `processing_time`
- `idle_time`

BUSINESS PROCESS ANALYTICS IN R

Let's practice!

BUSINESS PROCESS ANALYTICS IN R

Linking perspectives

Gert Janssenswillen
Creator of bupaR

Recap

	Organizational	Structuredness	Performance
Metrics	Resource Frequency Resource Specialization Resources Involvement	Start Activities End Activities Trace Coverage Trace Length Repetitions Self-loops Activity Presence	Processing Time Throughput Time Idle Time
Visuals	Resource Map	Process Map Trace Explorer Precedence Matrix	Performance Map Dotted Chart

Leveraging granularity levels

```
<process_metric>(level =     "log",
                             "trace",
                             "case",
                             "activity",
                             "resource",
                             "resource-activity")
```

- Performance <> Organizational

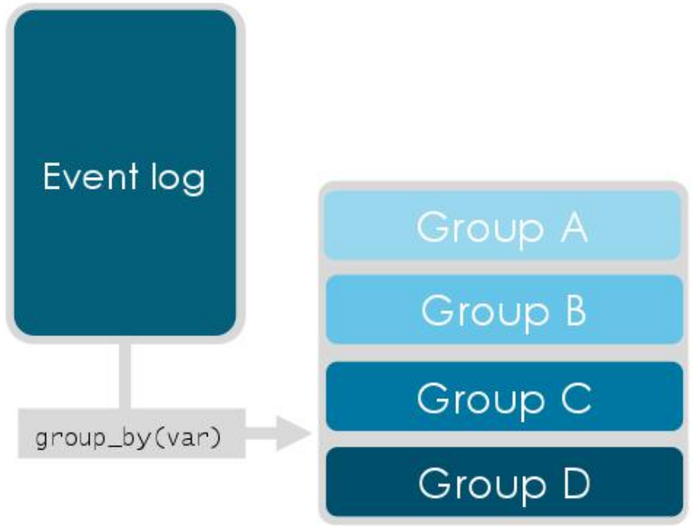
```
processing_time(level = "resource")
```

- Control-flow <> Organizational

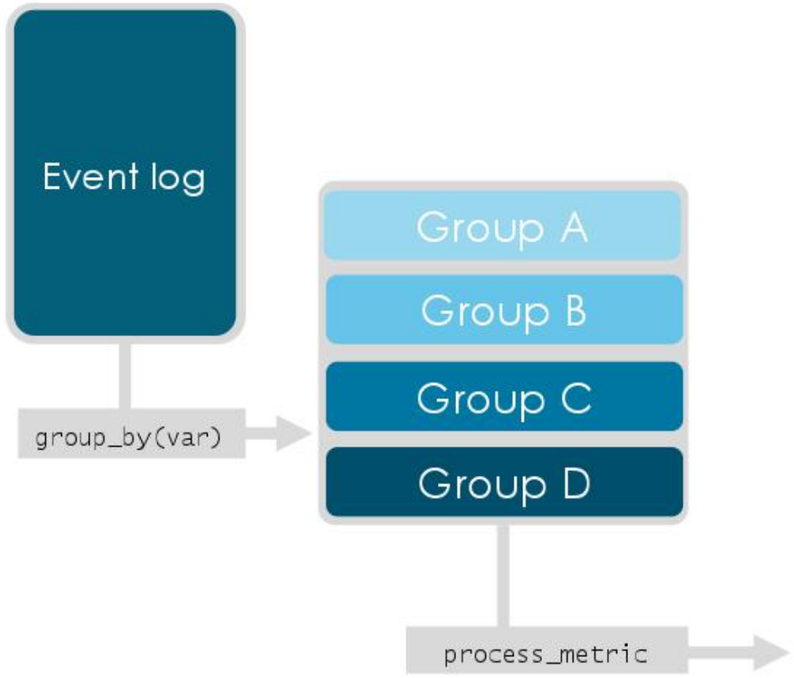
```
number_of_repetitions(level = "resource")
```

Grouping data

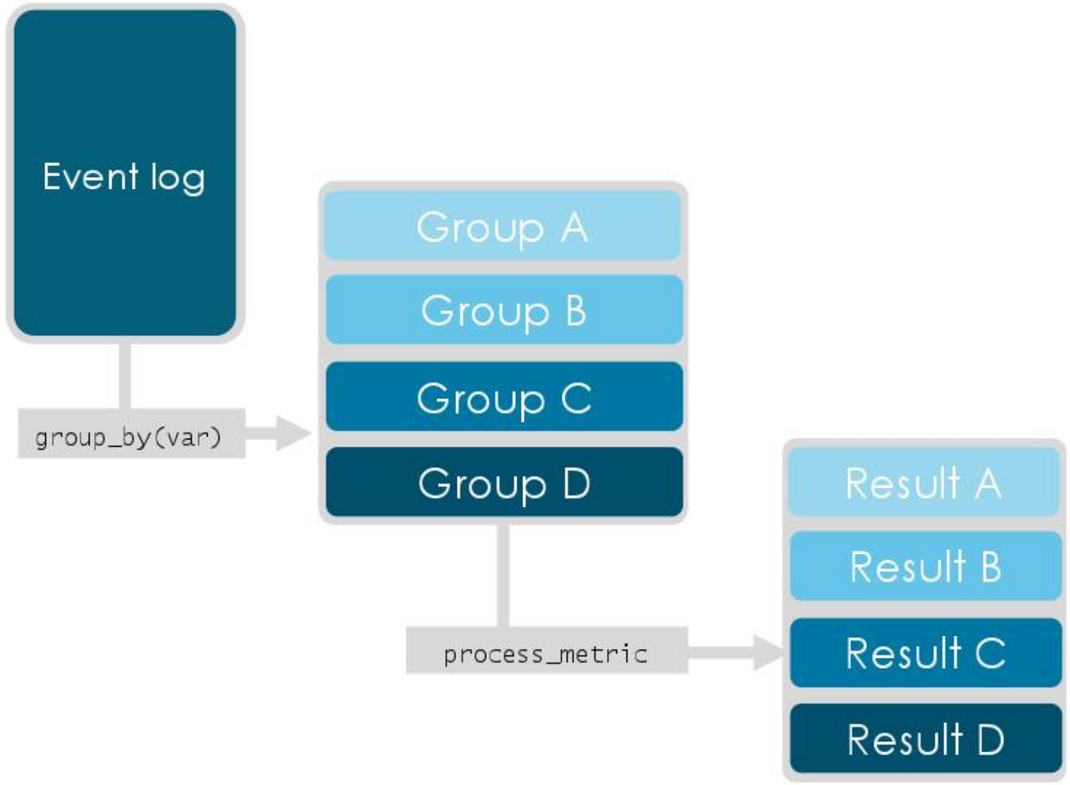
Grouping data



Grouping data



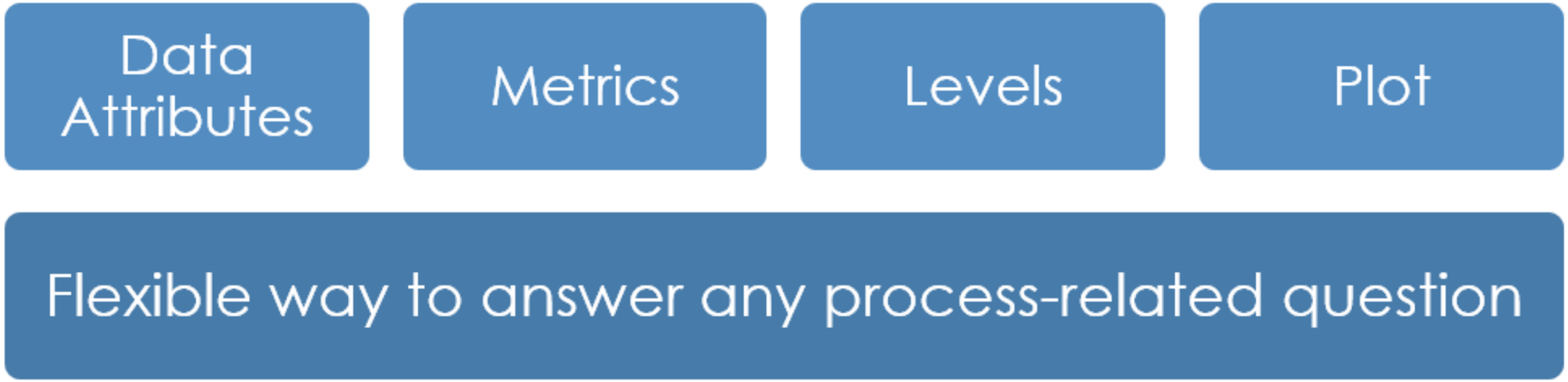
Grouping data



Grouping data: Example

Combining elements

Combining elements




```
eventlog %>%
    group_by(priority) %>%
    number_of_repetitions(level = "resource") %>%
    plot()
```

BUSINESS PROCESS ANALYTICS IN R

Let's practice!

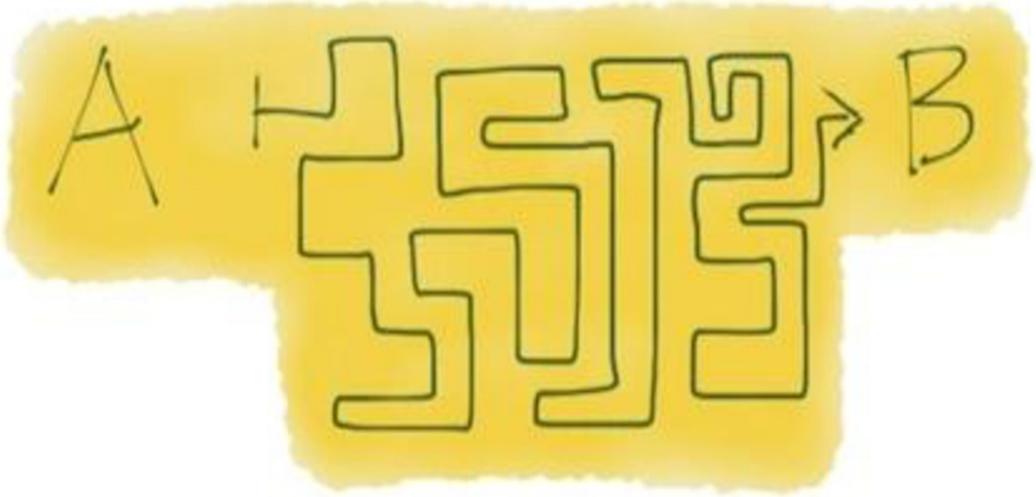
BUSINESS PROCESS ANALYTICS IN R

Filtering cases

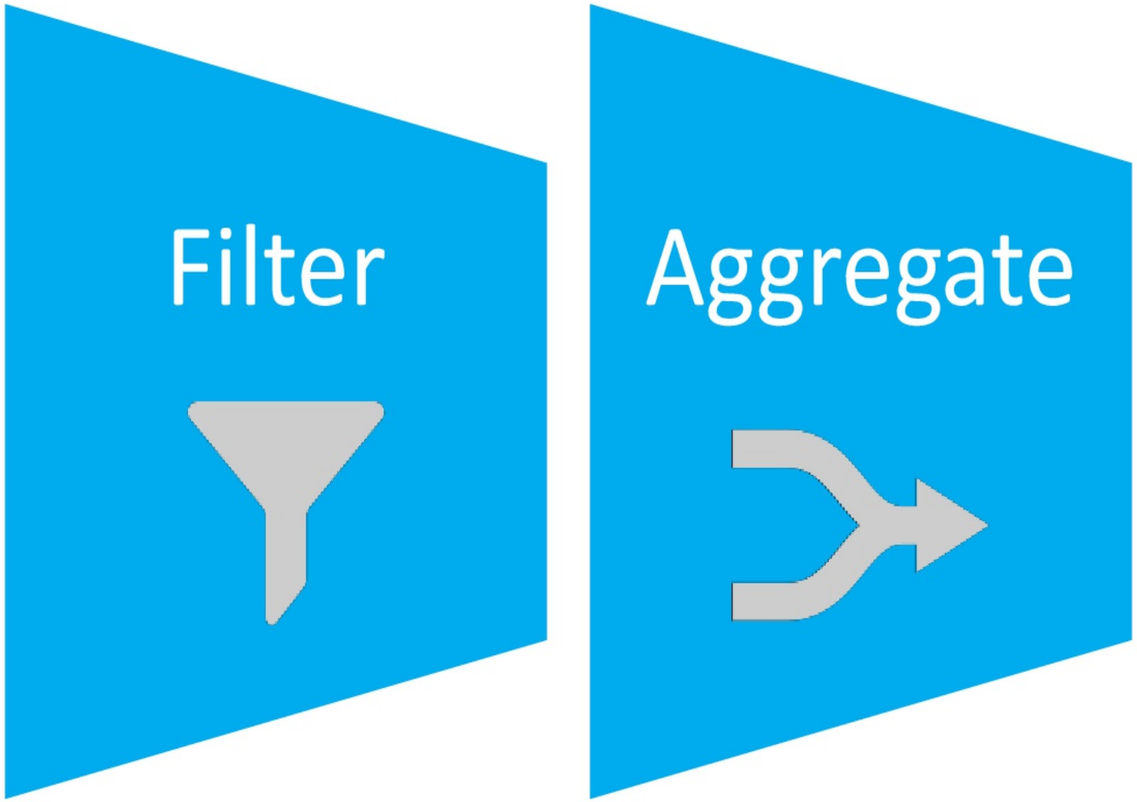
Gert Janssenswillen
Creator of bupaR

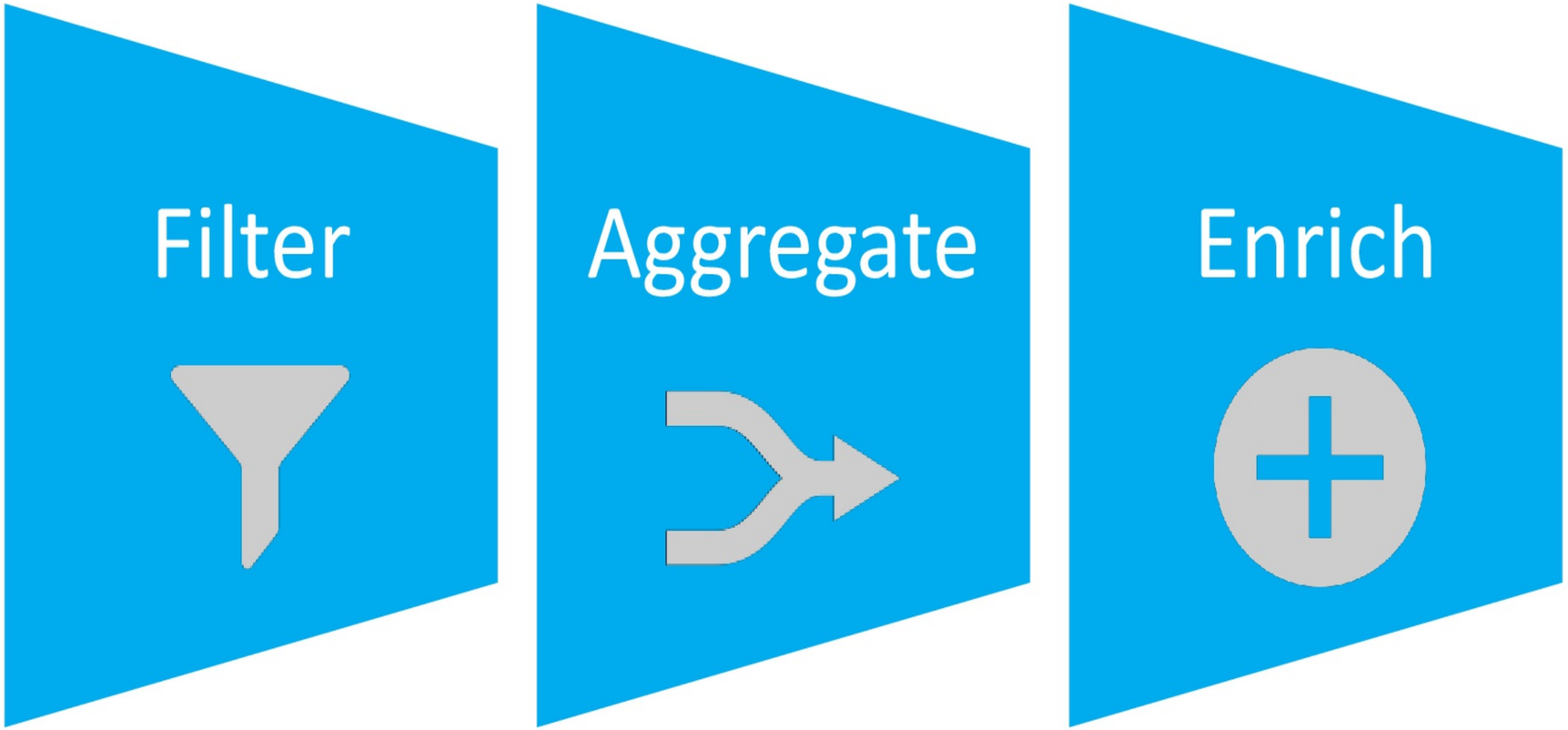
Theory

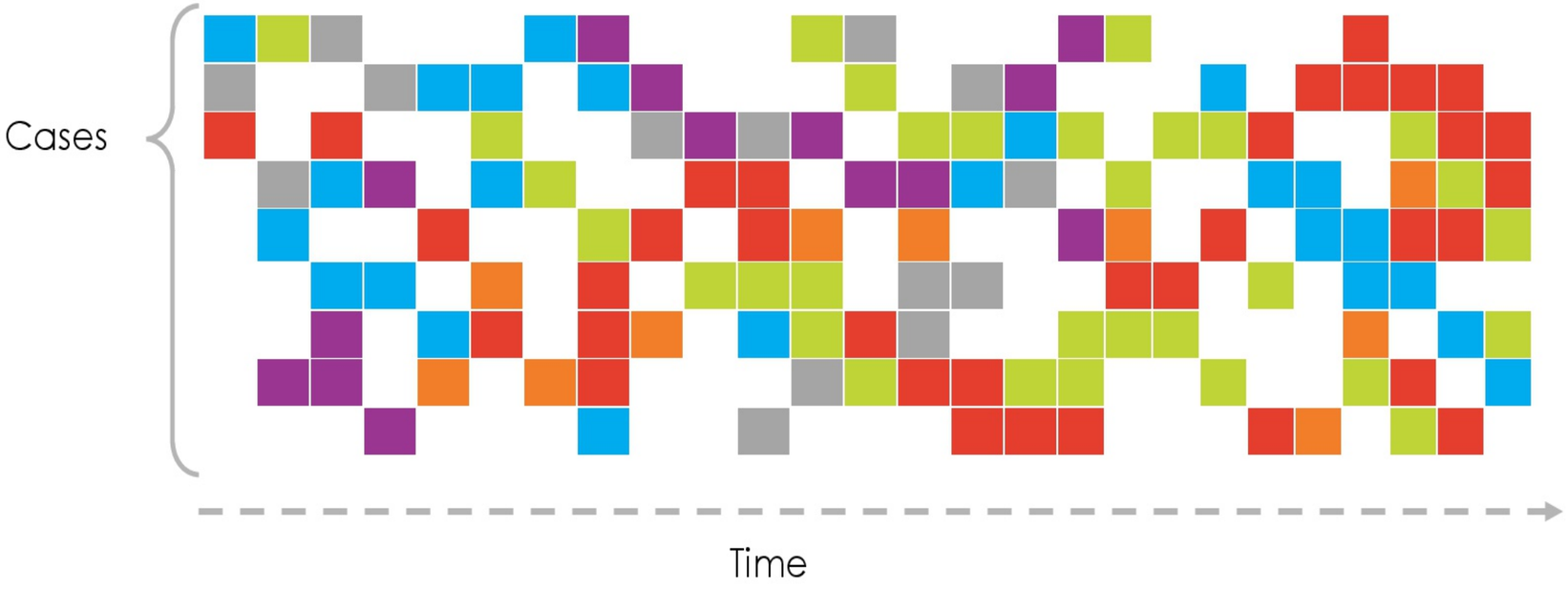
Practice

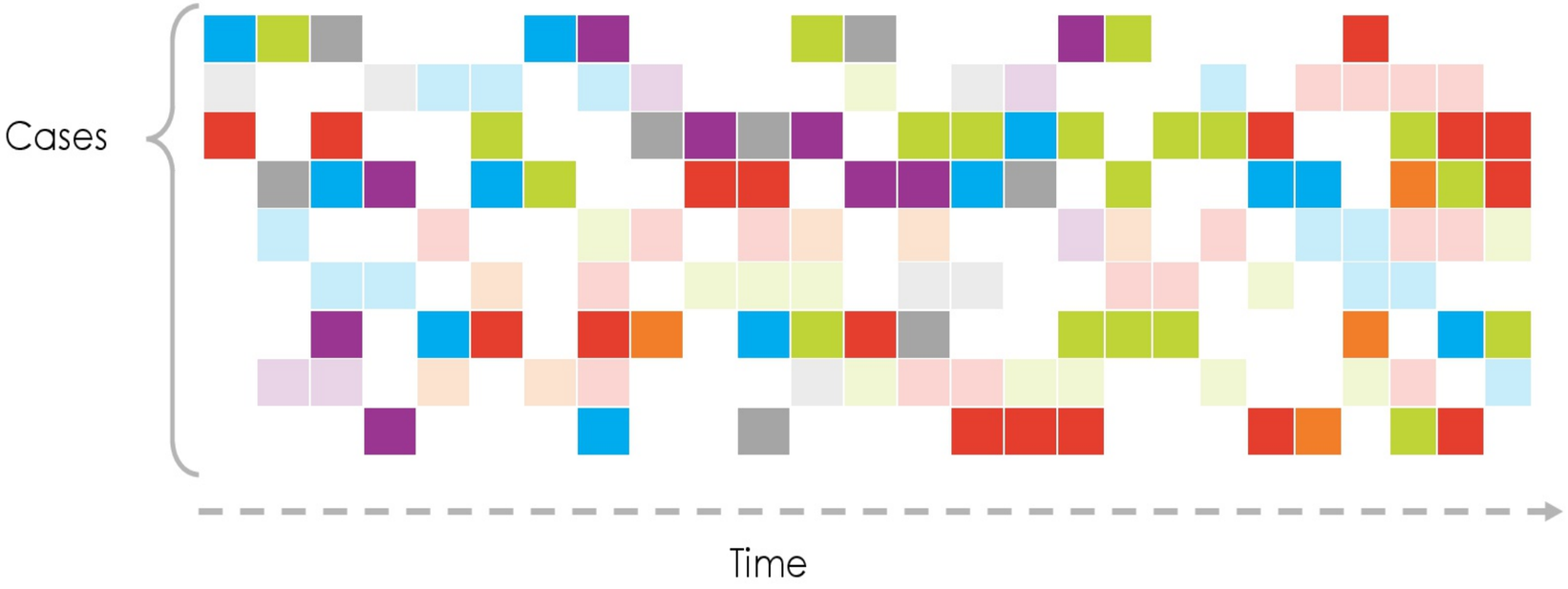


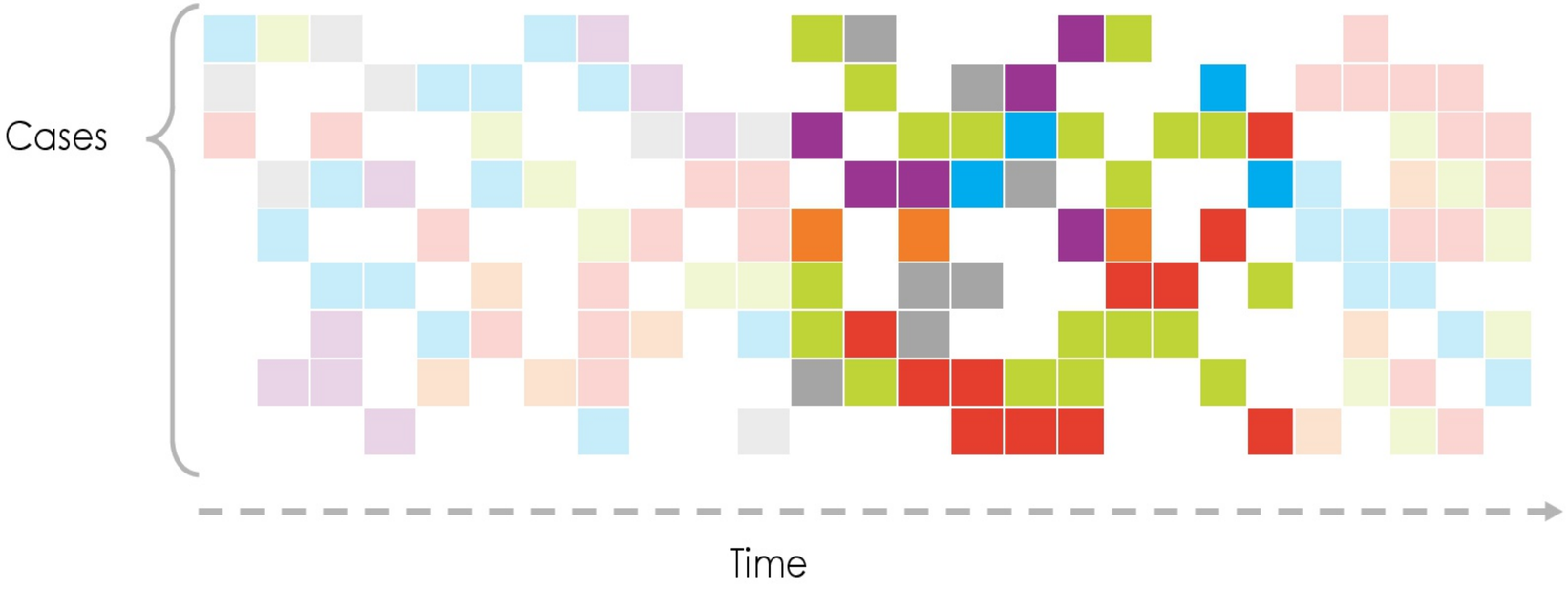

Filter

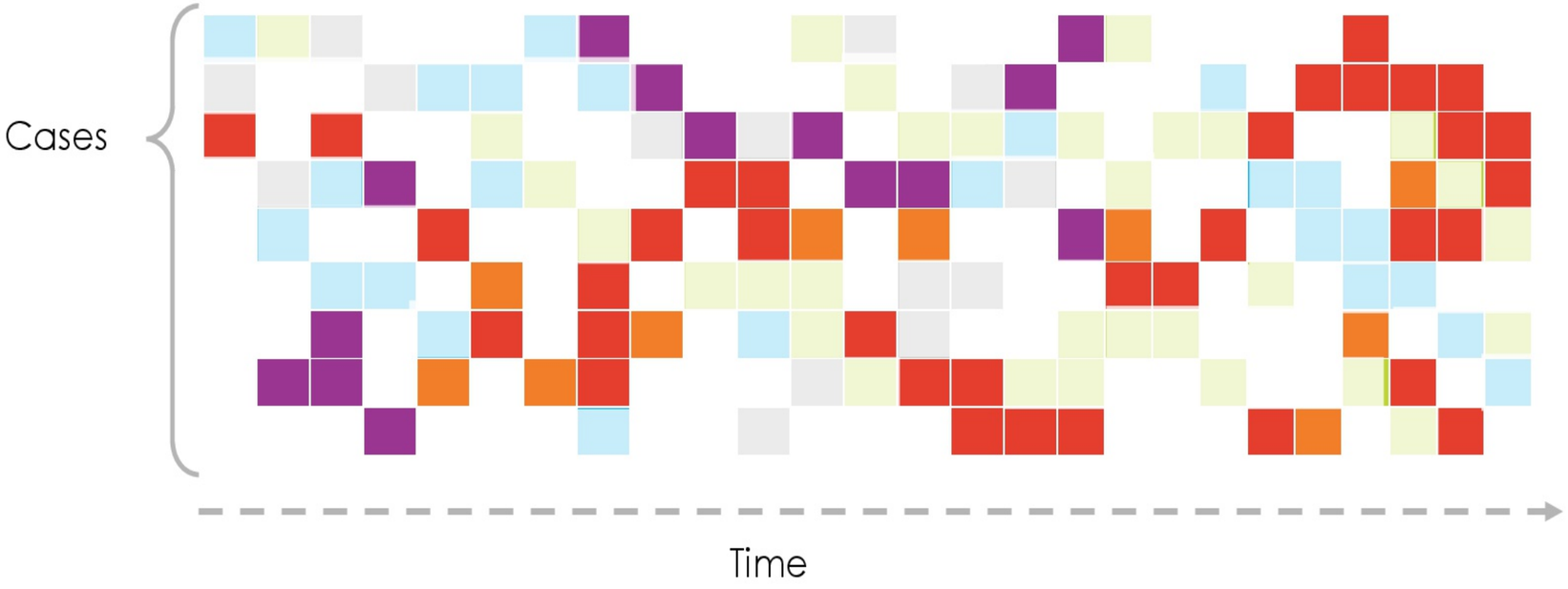

Filter
Aggregate


Filter
Aggregate
Enrich


Cases
Time


Cases
Time


Cases
Time

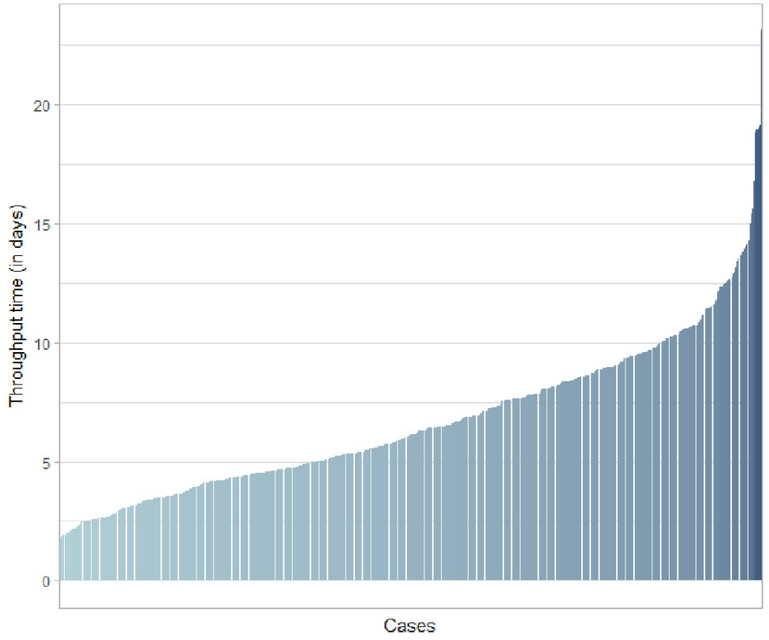

Cases
Time

Categories of Case Filters

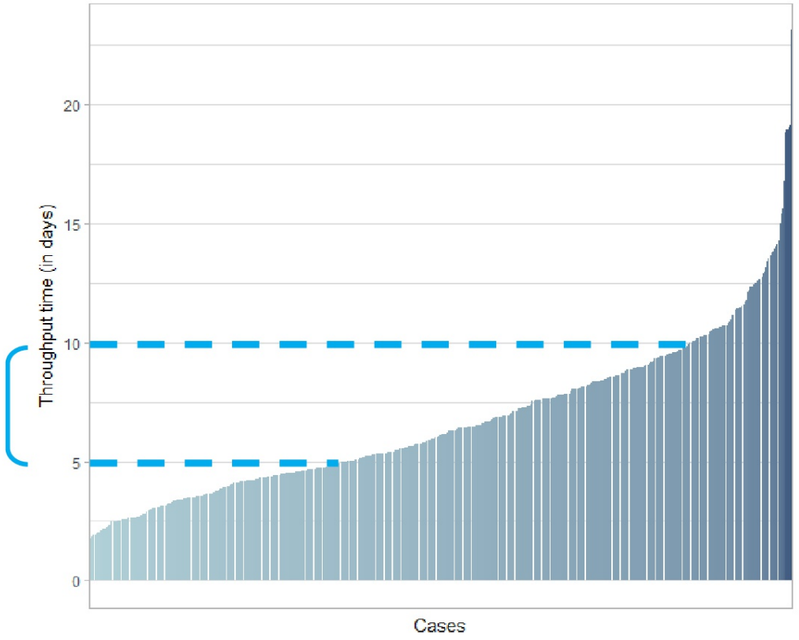
- Performance
- Control-flow characteristics
- Time period

Performance filters

Performance filters

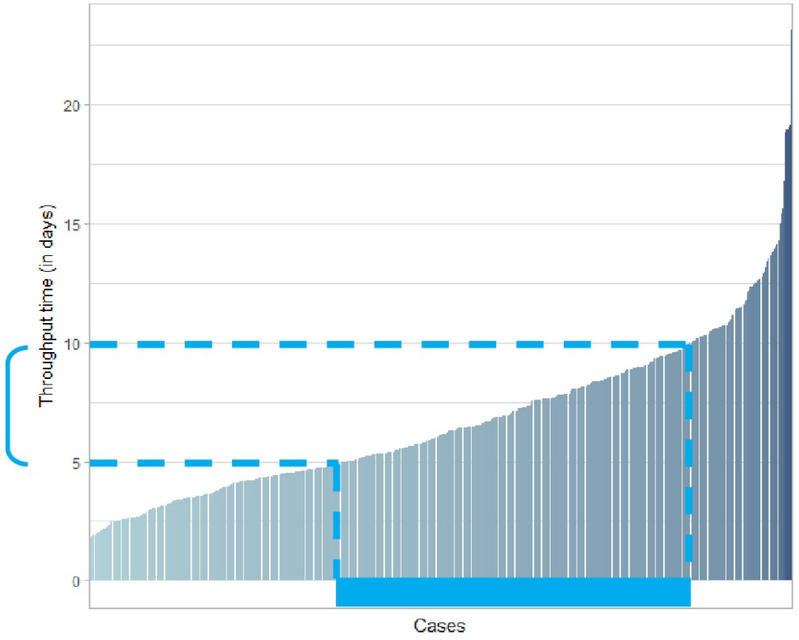


Filter by absolute interval



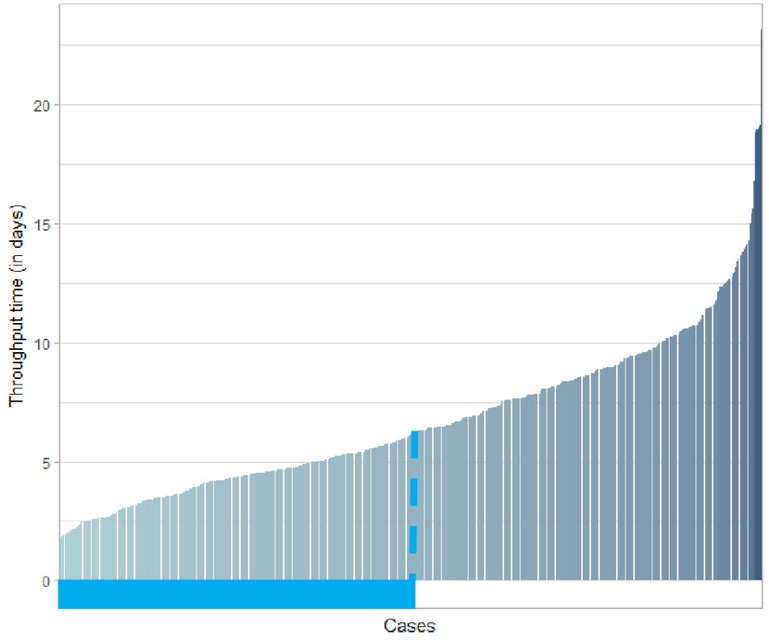

```
filter_throughput_time(log, interval = c(5,10))
```

Filter by absolute interval




```
filter_throughput_time(log, interval = c(5,10))
```

Filter by Relative Percentage




```
filter_throughput_time(log, percentage = 0.5)
```

Adjusting filter configurations

Negate the filter

- Cases shorter than 5 days, or longer than 10 days

```
filter_throughput_time(log, interval = c(5,10), units = "days", reverse =TRUE)
```

- The 50% longest cases

```
filter_throughput_time(log, percentage = 0.5, reverse = TRUE)
```

Use half-open intervals

- Select cases with throughput time longer than 5 days.

```
filter_throughput_time(log, interval = c(5,NA), units = "days")
```

Control-flow filters

- Activity presence/absence
- Precendence requirements
- Start and/or End points
- Frequency of the trace

Time filters

Select cases that

- **started** in a specific time window
- **ended** in a specific time window
- are **contained** in a specific time window
- **intersect**, i.e. had at least on activity in a specific time window

BUSINESS PROCESS ANALYTICS IN R

Let's practice!

BUSINESS PROCESS ANALYTICS IN R

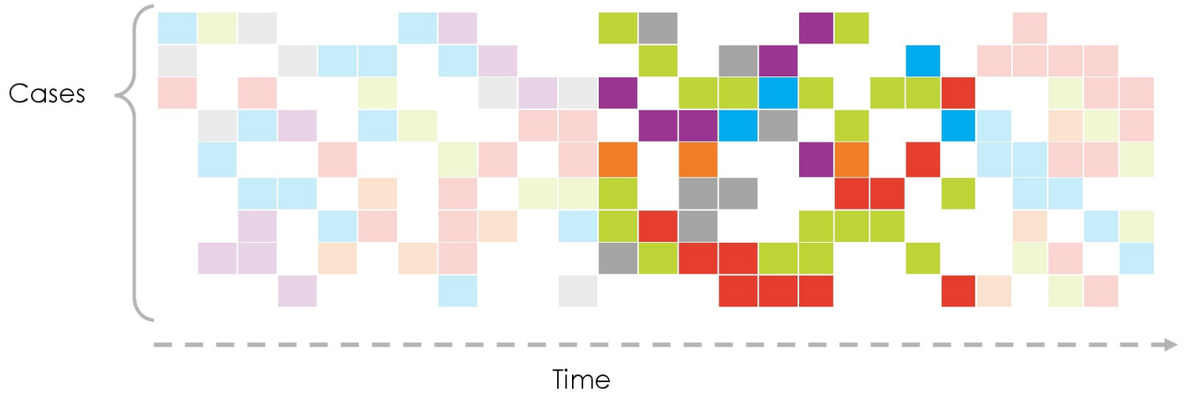
Filtering events

Gert Janssenswillen
Creator of bupaR

Categories of event filters

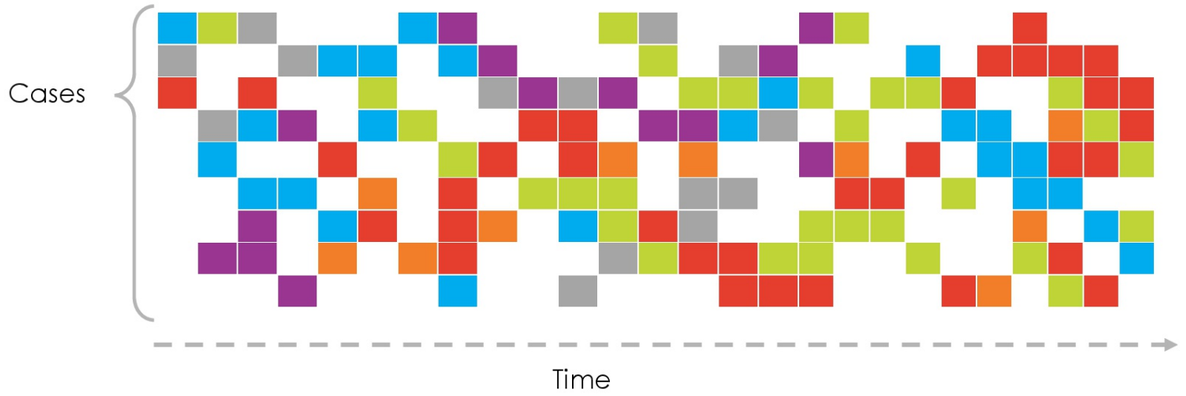
- Trim filters
- Frequency filters
- Label filters
- General Attribute filters

Trim to time period



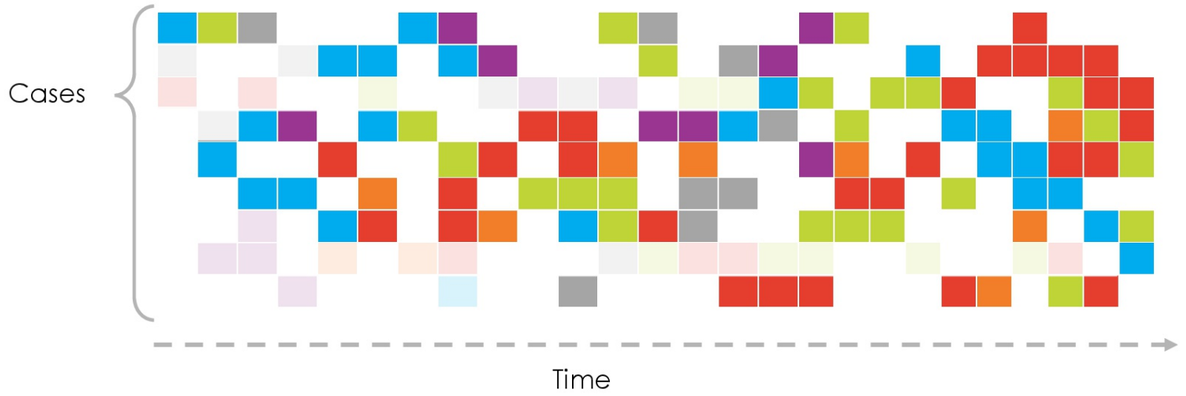

```
filter_time_period(log, interval = ymd(c("20180110","20180122")),
                   filter_method = "trim")
```

Trim to start and end points



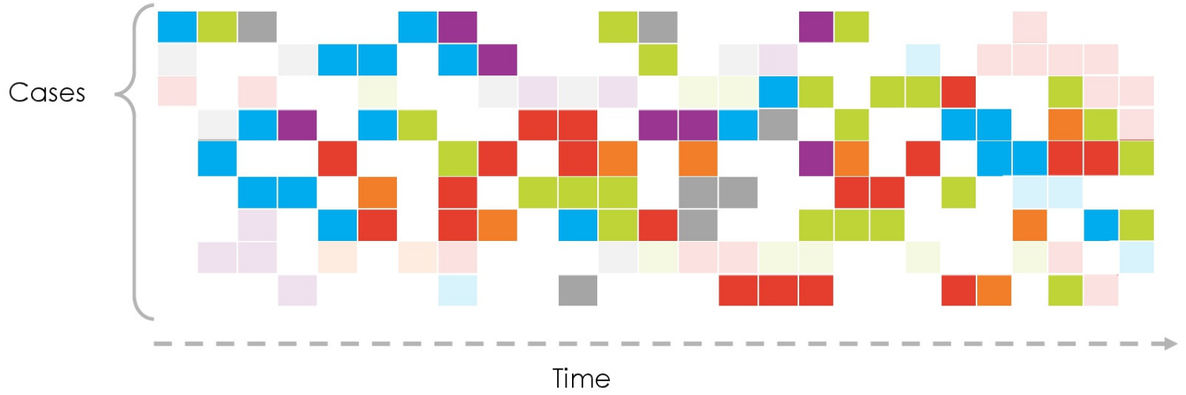

```
filter_trim(start_activities = "blues")
```

Trim to start and end points




```
filter_trim(start_activities = "blues", end_activities = "greens")
```

Trim to start and end points



Filter by frequencies

- Activity frequency

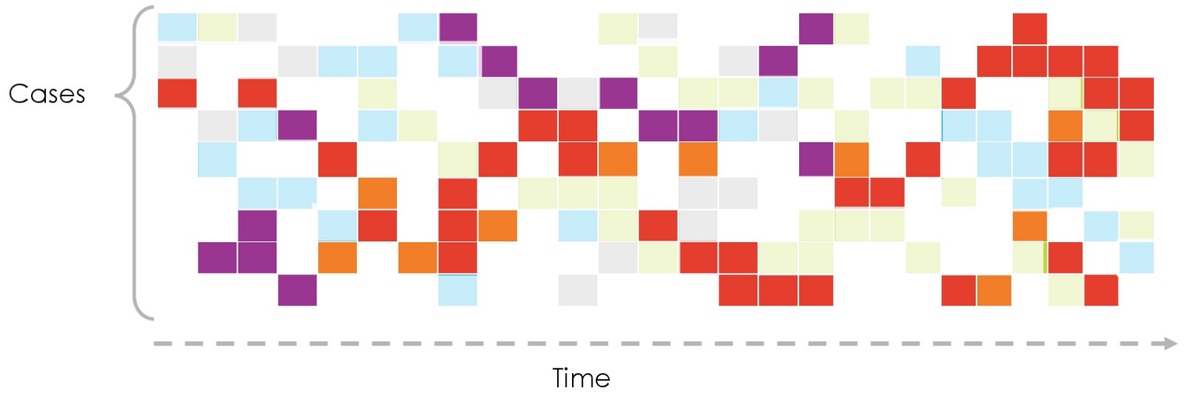
```
filter_activity_frequency(log, interval = c(50,100))
filter_activity_frequency(log, percentage = 0.8)
```

- Resource frequency

```
filter_resource_frequency(log, interval = c(60,900))
filter_resource_frequency(log, percentage = 0.6)
```

Filter by labels

```
filter_activity(log, activities = c("reds","oranges","purples")))
```



Filter by conditions

```
filter(log, cost > 1000, priority == "High", ...)
```

- Any condition using data attributes can be used
- Multiple conditions can be combined using &, |, !, etc.

BUSINESS PROCESS ANALYTICS IN R

Let's practice!

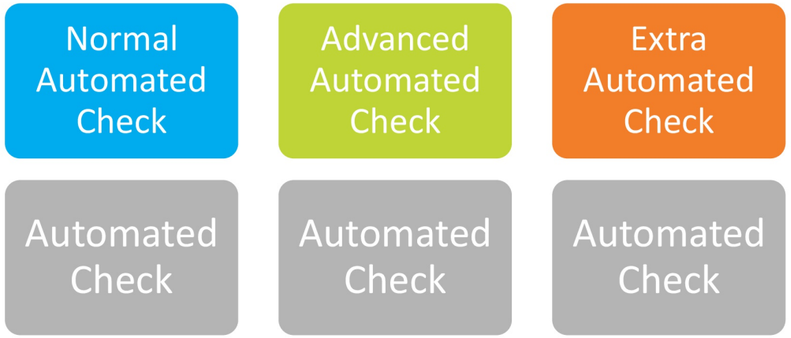
BUSINESS PROCESS ANALYTICS IN R

Aggregating events

Gert Janssenswillen
Creator of bupaR

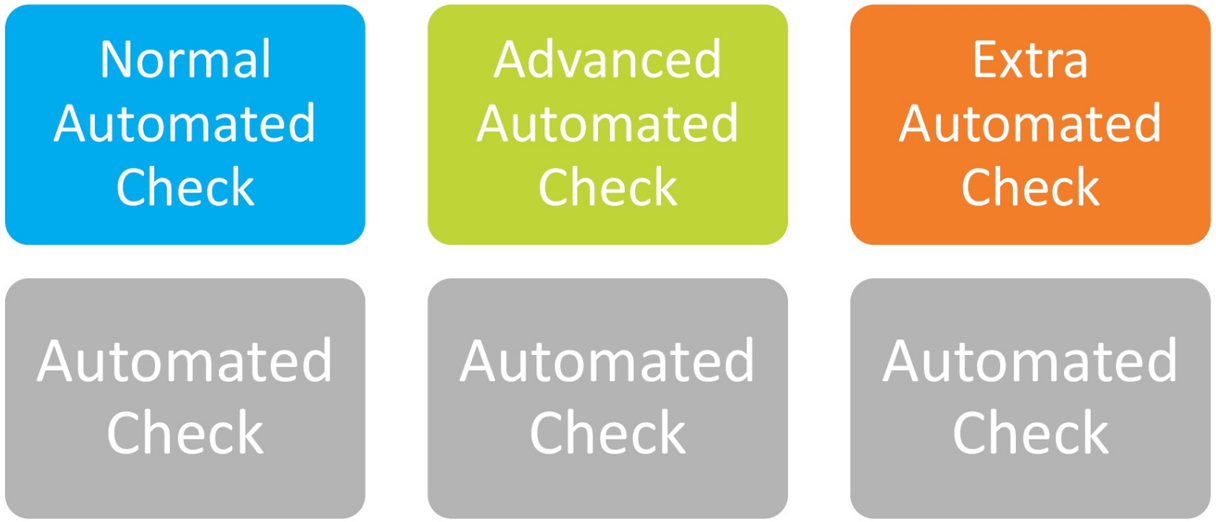
Aggregation types

Is-A



Part-of

Is-a aggregation



Is-a aggregation

```
act_unite(log, "New name" = c("Old Variant 1","Old Variant 2","Old Variant 3"),
          ...)
```

Part-of aggregation

Part-of aggregation

```
act_collapse(log, "Sub process" = c("Part 1","Part 2","Part 3"),
             ...)
```

Impact on Activity Types and Instances

Is-a

- Decreased number of activity **types**
- Equal number of activity **instances**

Part-of

- Decreased number of activity **types**
- Decreased number of activity **instances**

BUSINESS PROCESS ANALYTICS IN R

Let's practice!

BUSINESS PROCESS ANALYTICS IN R

Enriching events

Gert Janssenswillen
Creator of bupaR

Mutate new variables

Mutate new variables

```
log %>%
    group_by_case() %>%
    mutate(total_cost = sum(cost, na.rm = T)
```

Mutate new variables

```
log %>%
    group_by_case %>%
    mutate(total_cost = sum(cost, na.rm = T) %>%
    mutate(impact = case_when(cost <= 1000 ~ "Low",
                              cost <= 5000 ~ "Medium",
                              T ~ "High"))
```

Mutate new variables

```
log %>%
    group_by_case %>%
    mutate(refund_made = any(str_detect(activity, "Pay Claim")))
```

Adding process metrics

Adding information about a case to the original data

- Its througput time
- Its length
- Its amount of rework
- ...

Adding information about activities

- Its frequency
- Its specialization by resources
- ...

Adding process metrics

```
log %>%
    throughput_time(level = "case", units = "days", append = TRUE)
```

```
log %>%
    throughput_time(level = "case", units = "days", append = TRUE) %>%
    mutate(on_time = processing_time_case <= 7)
```

BUSINESS PROCESS ANALYTICS IN R

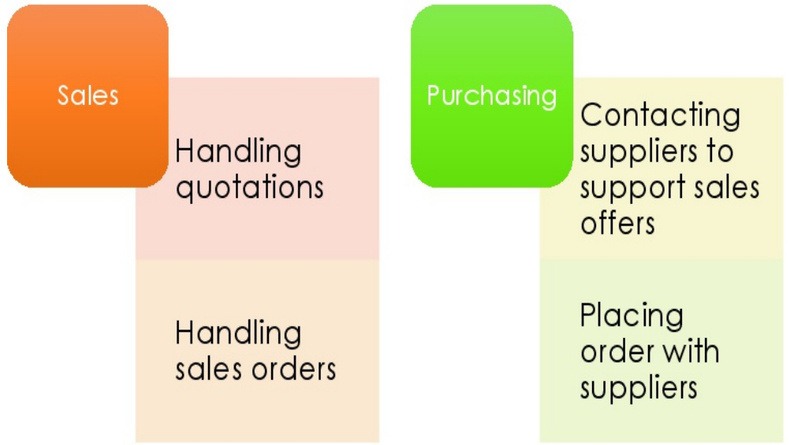
Let's practice!

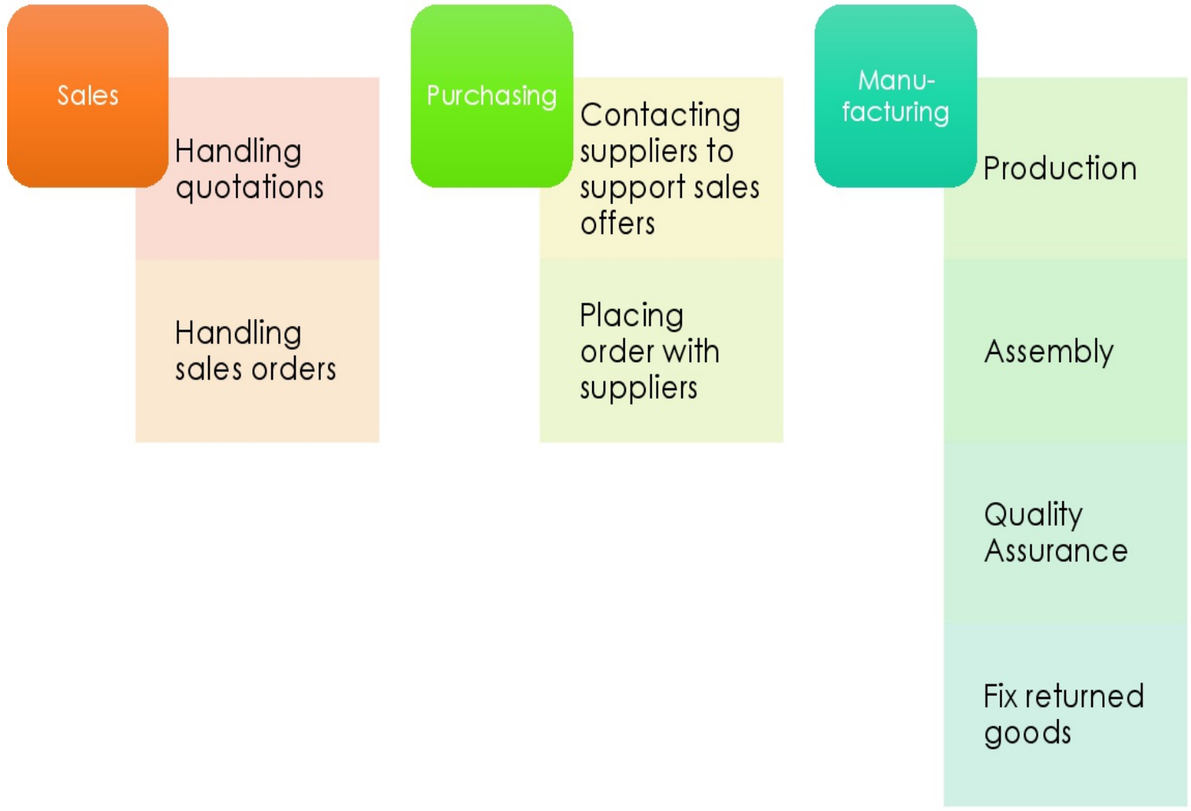
BUSINESS PROCESS ANALYTICS IN R

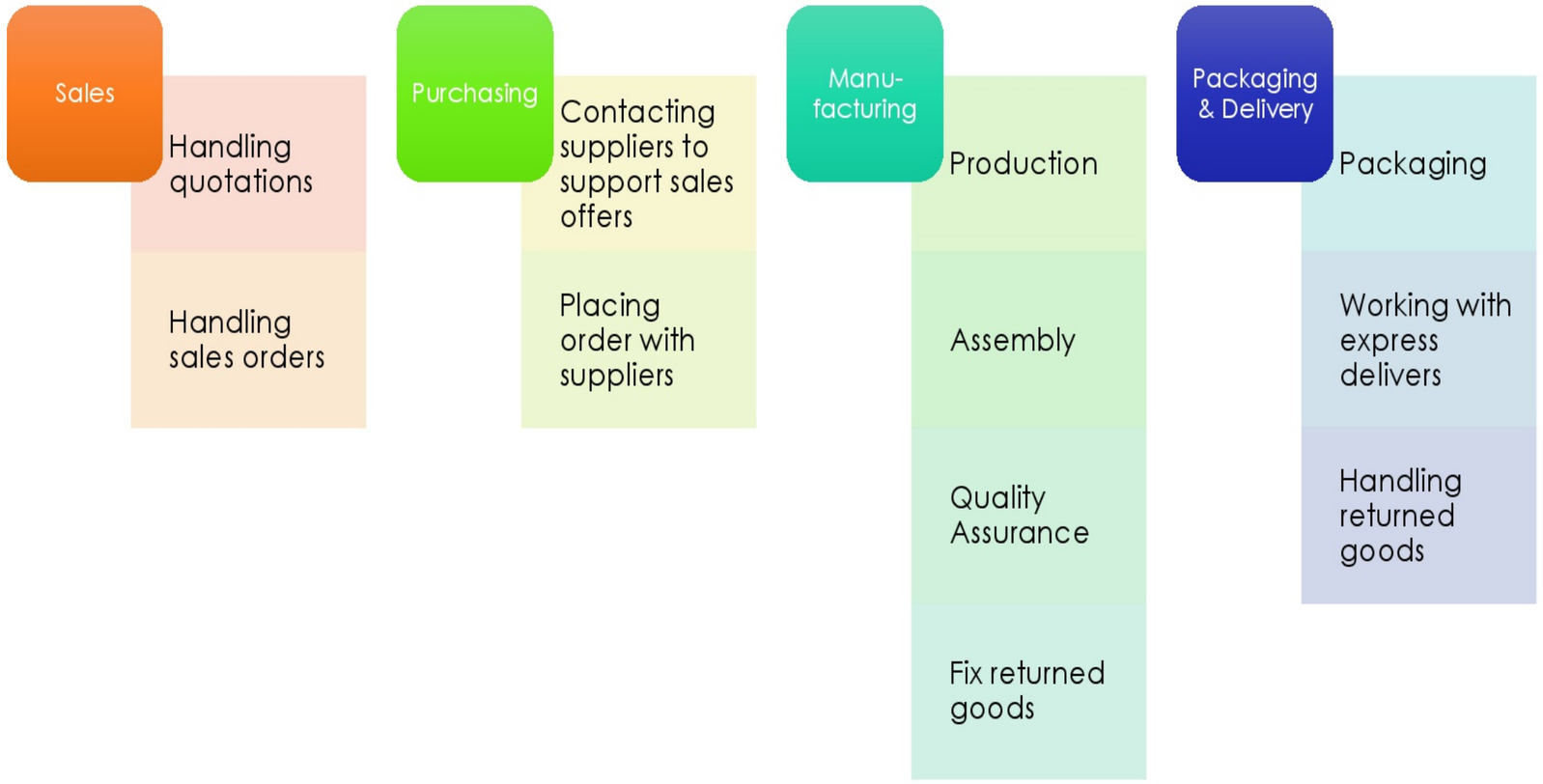
Preparing the event data

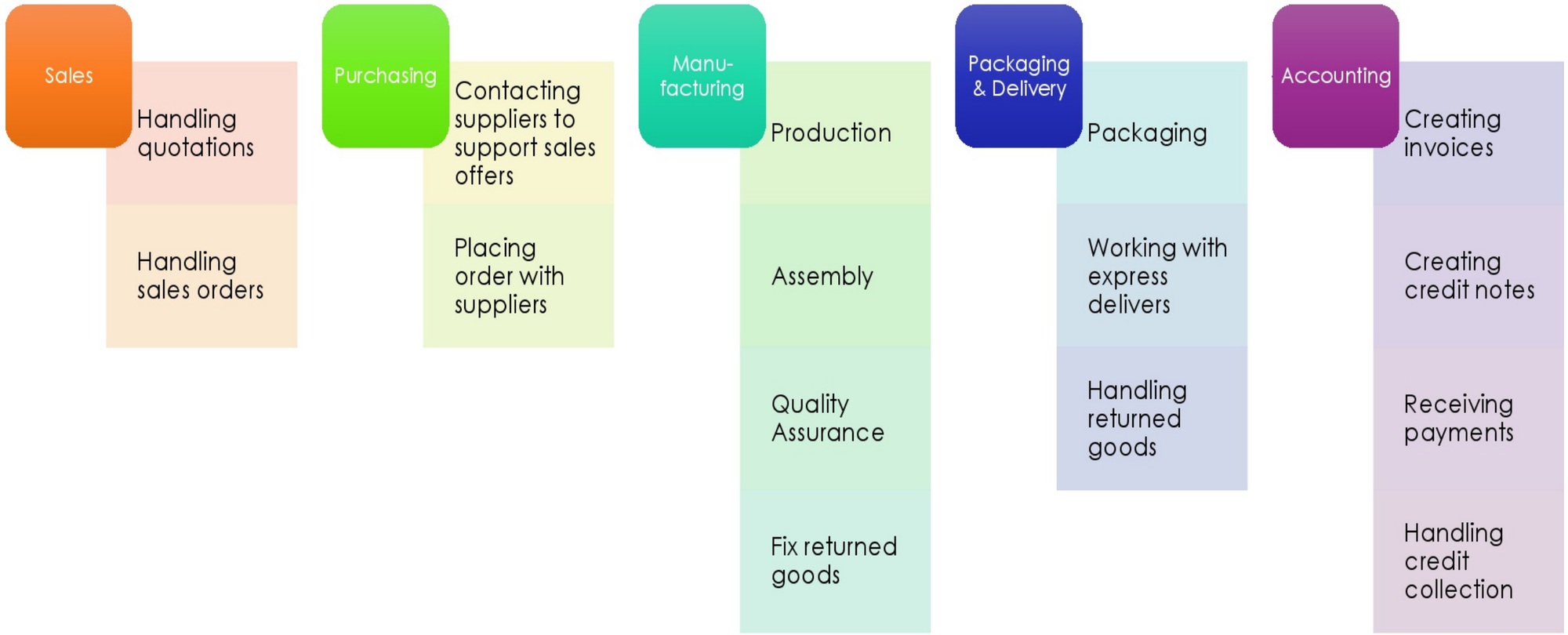
Gert Janssenswillen
Creator of bupaR

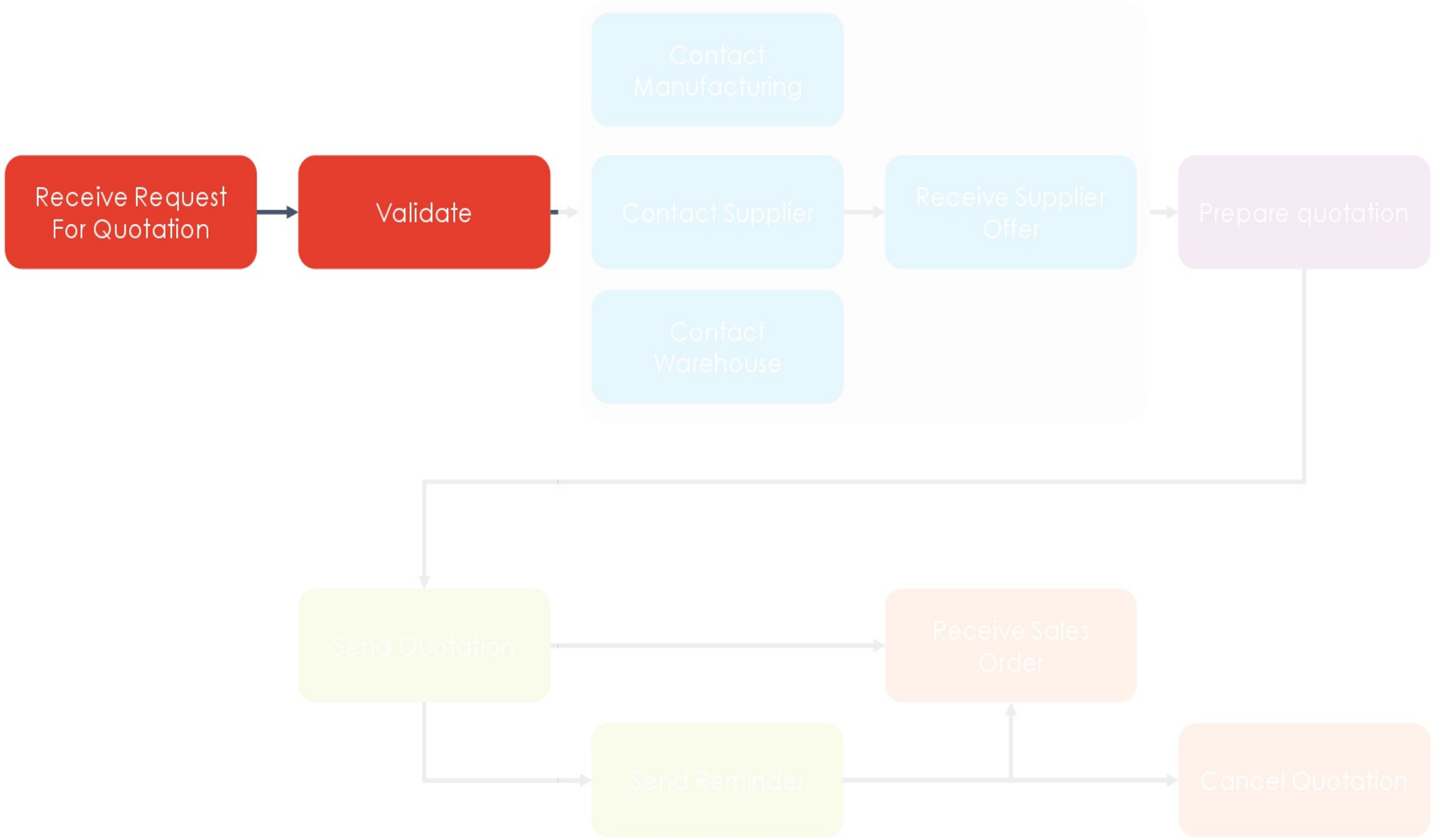

Sales
Handling quotations
Handling sales orders

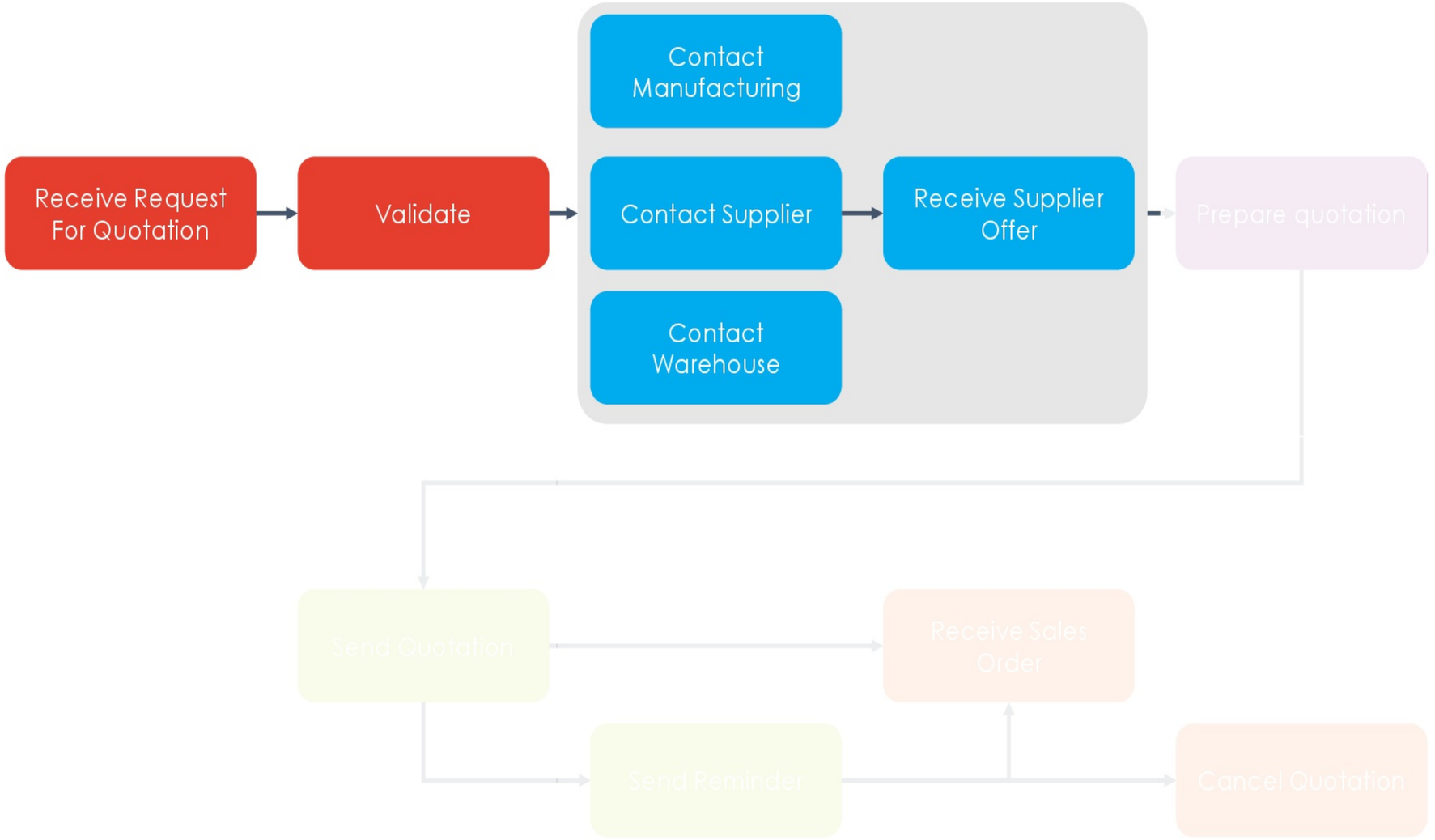

Sales
Handling quotations
Handling sales orders
Purchasing
Contacting suppliers to support sales offers
Placing order with suppliers

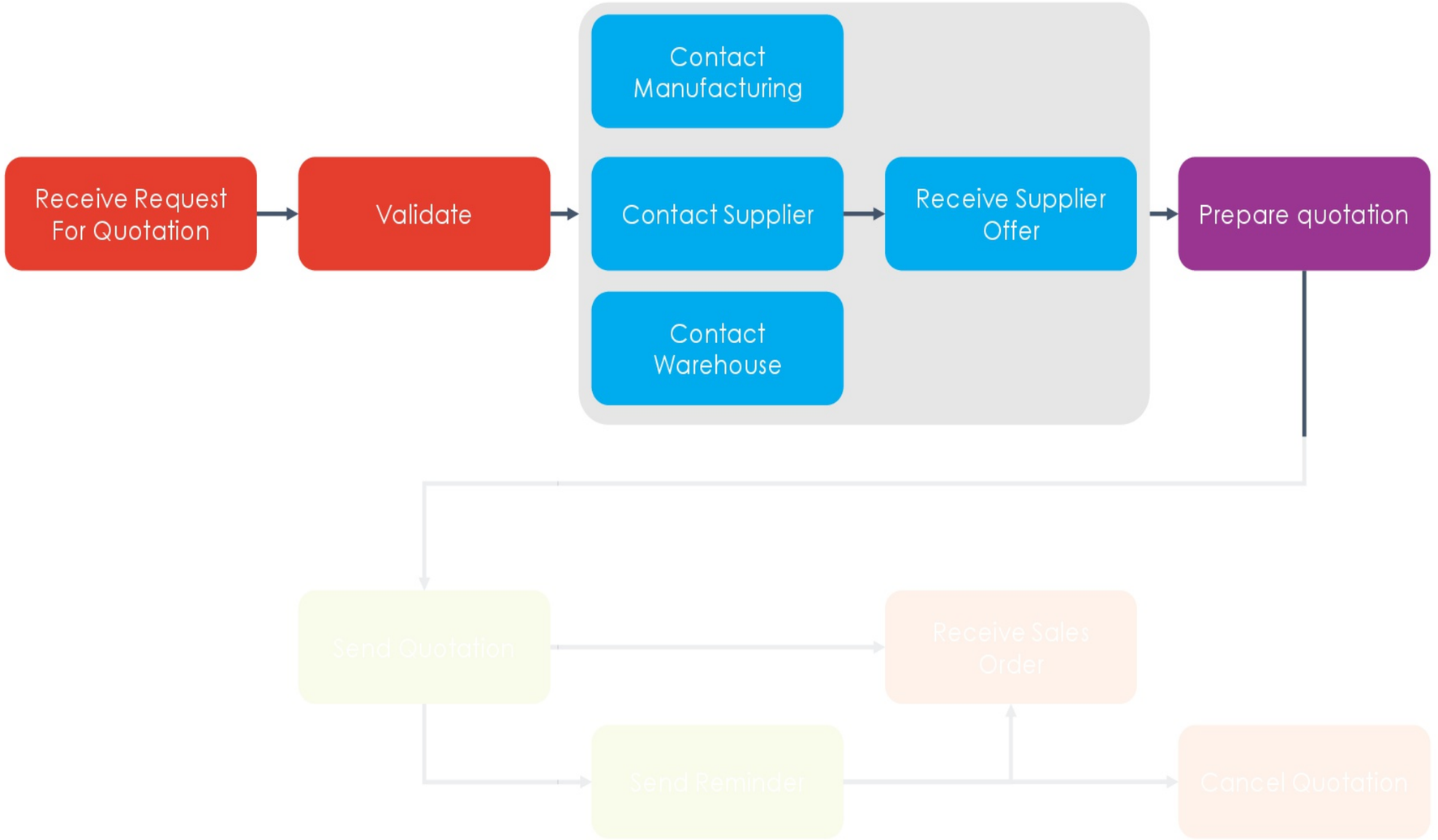

Sales
Handling quotations
Handling sales orders
Purchasing
Contacting suppliers to support sales offers
Placing order with suppliers
Manu-facturing
Production
Assembly
Quality Assurance
Fix returned goods

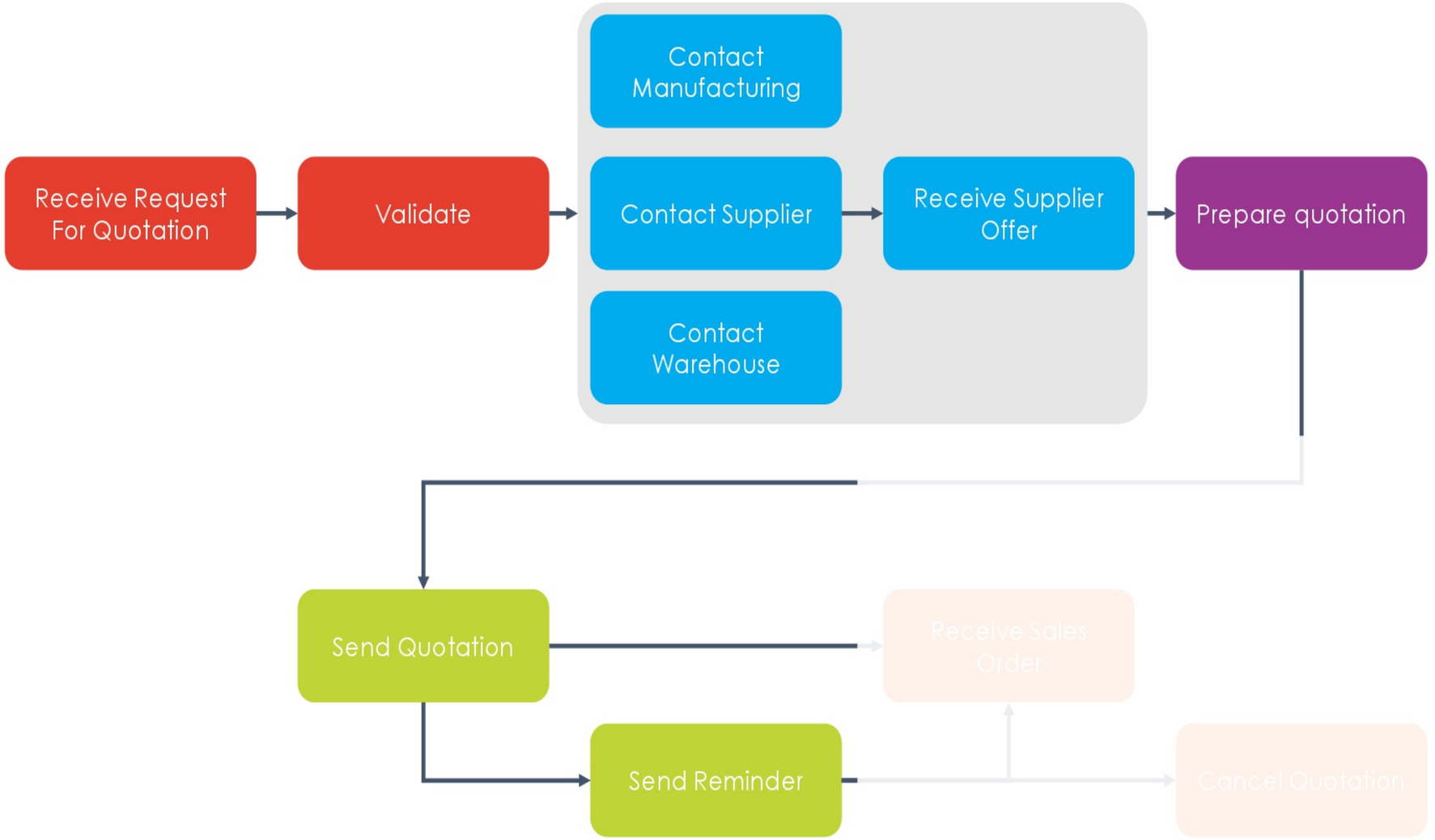

Sales
Handling quotations
Handling sales orders
Purchasing
Contacting suppliers to support sales offers
Placing order with suppliers
Manu-facturing
Production
Assembly
Quality Assurance
Fix returned goods
Packaging & Delivery
Packaging
Working with express delivers
Handling returned goods

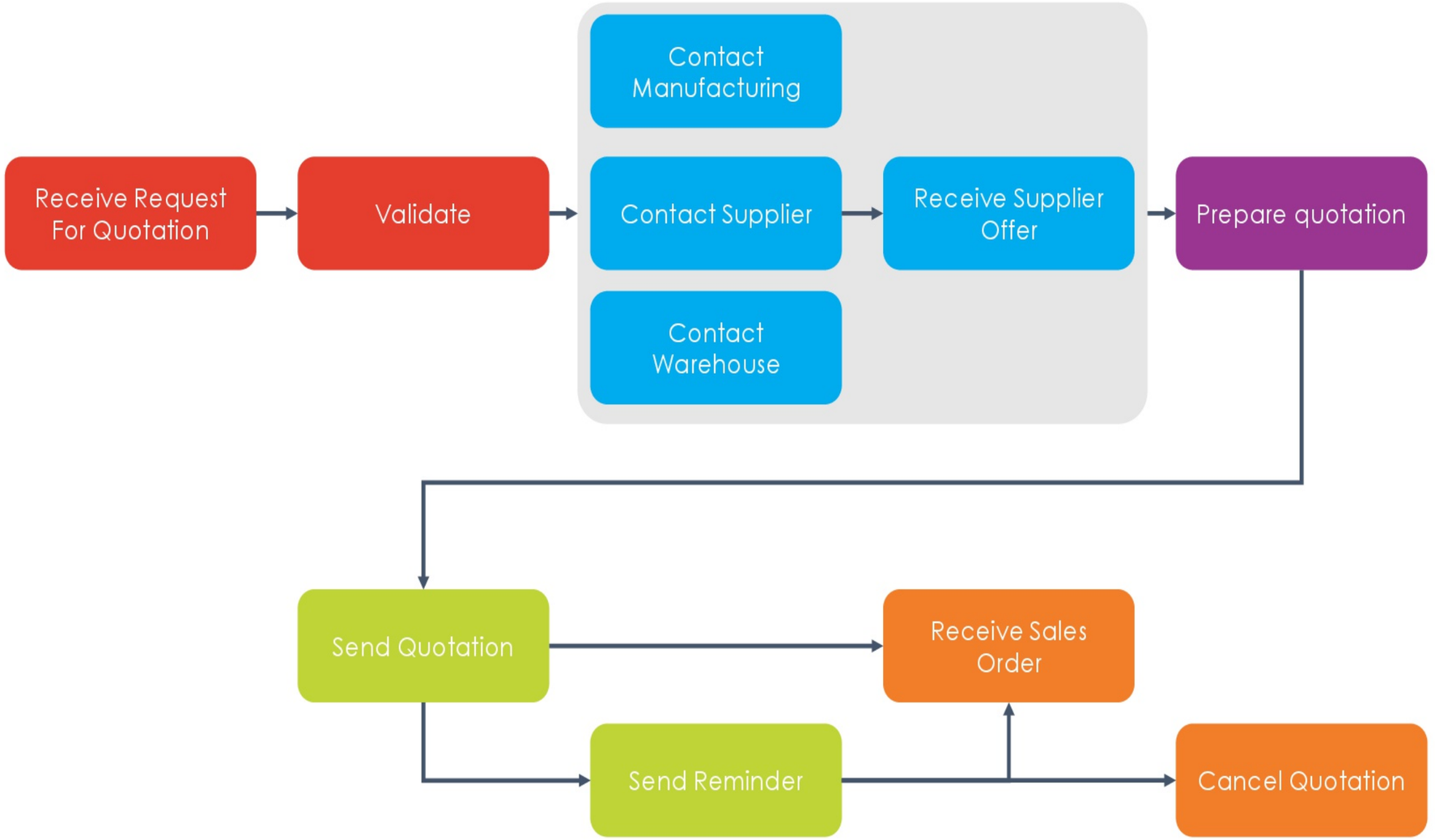

Sales
Handling quotations
Handling sales orders
Purchasing
Contacting suppliers to support sales offers
Placing order with suppliers
Manu-facturing
Production
Assembly
Quality Assurance
Fix returned goods
Packaging & Delivery
Packaging
Working with express delivers
Handling returned goods
Accounting
Creating invoices
Creating credit notes
Receiving payments
Handling credit collection


Receive Request For Quotation
Validate
Contact Manufacturing
Contact Supplier
Contact Warehouse
Receive Supplier Offer
Prepare quotation
Send Quotation
Receive Sales Order
Send Reminder
Cancel Quotation

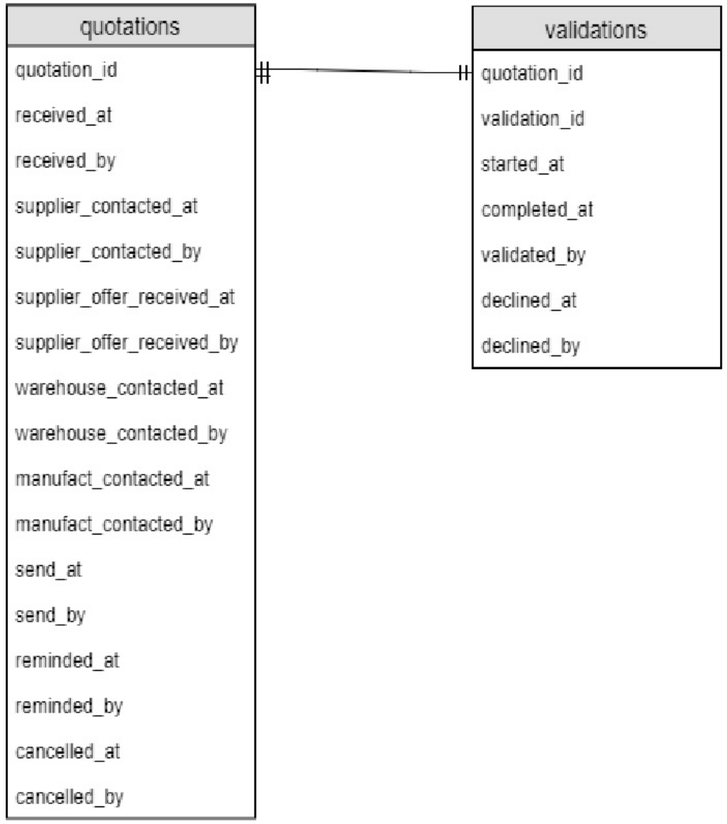

Receive Request For Quotation
Validate
Contact Manufacturing
Contact Supplier
Contact Warehouse
Receive Supplier Offer
Prepare quotation
Send Quotation
Receive Sales Order
Send Reminder
Cancel Quotation

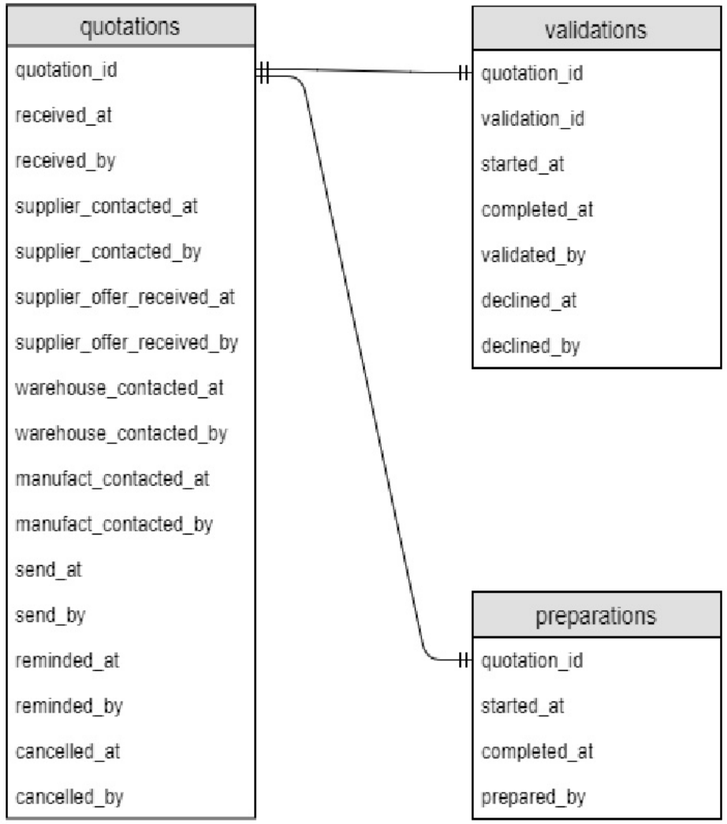

Contact Manufacturing
Receive Request For Quotation
Validate
Contact Supplier
Receive Supplier Offer
Prepare quotation
Contact Warehouse
Send Quotation
Receive Sales Order
Send Reminder
Cancel Quotation

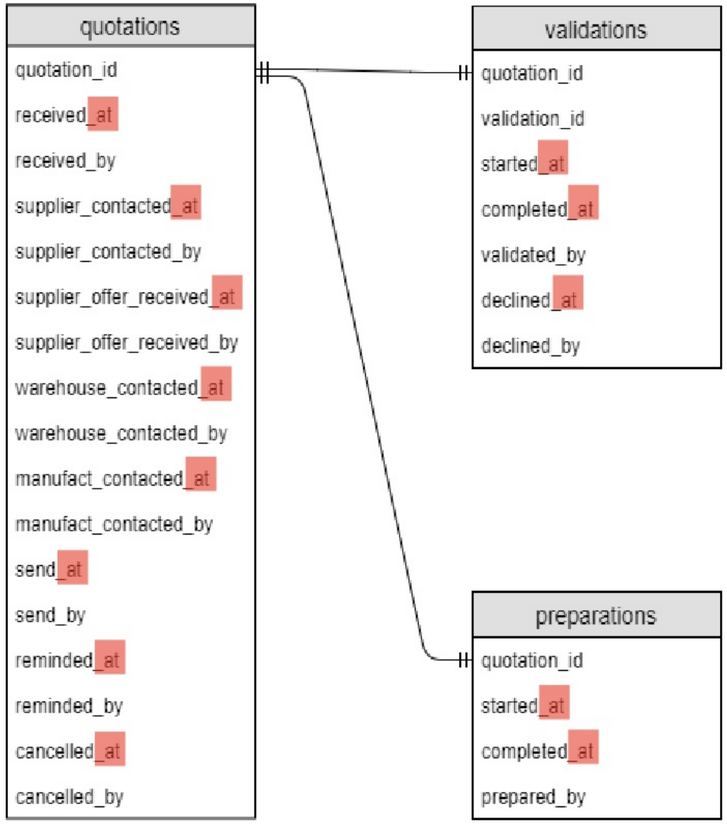

Receive Request For Quotation
Validate
Contact Manufacturing
Contact Supplier
Contact Warehouse
Receive Supplier Offer
Prepare quotation
Send Quotation
Receive Sales Order
Send Reminder
Cancel Quotation

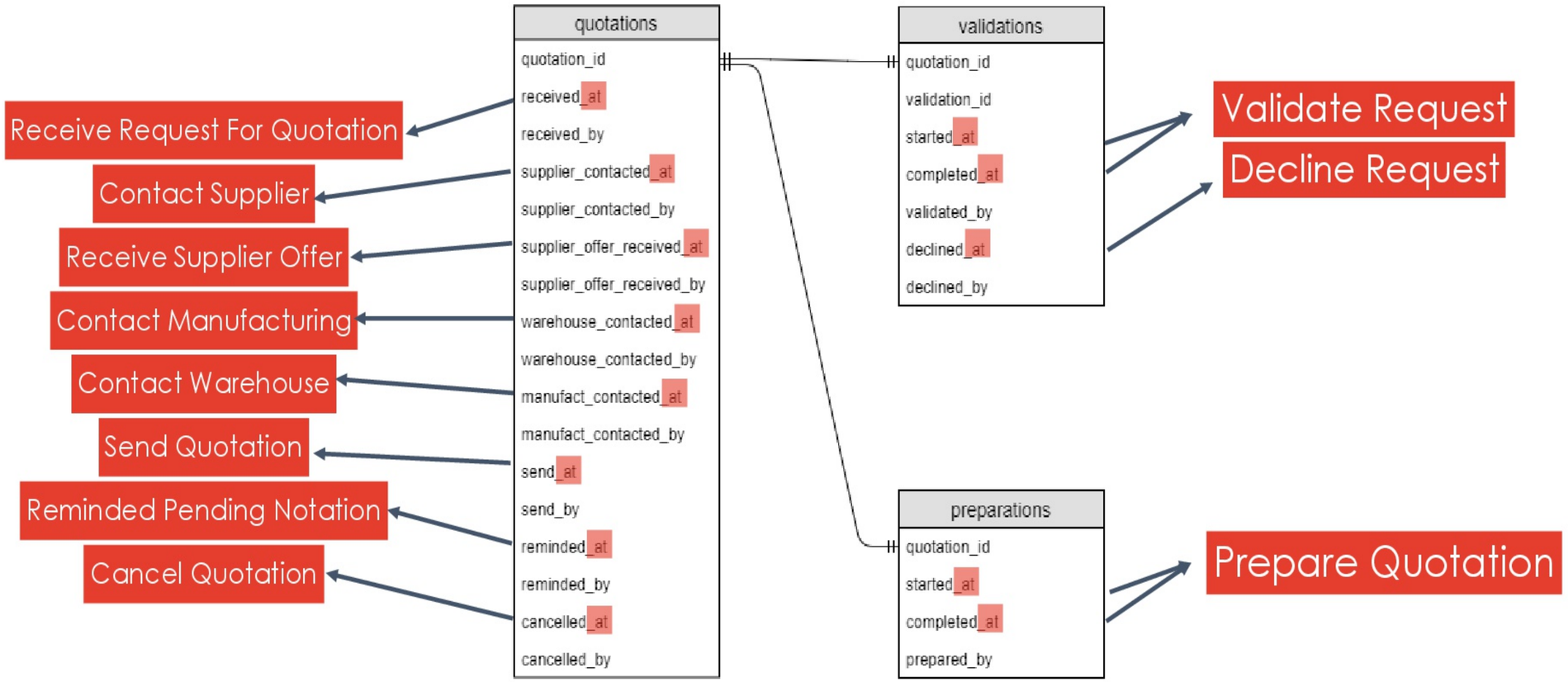

Receive Request For Quotation
Validate
Contact Manufacturing
Contact Supplier
Contact Warehouse
Receive Supplier Offer
Prepare quotation
Send Quotation
Receive Sales Order
Send Reminder
Cancel Quotation

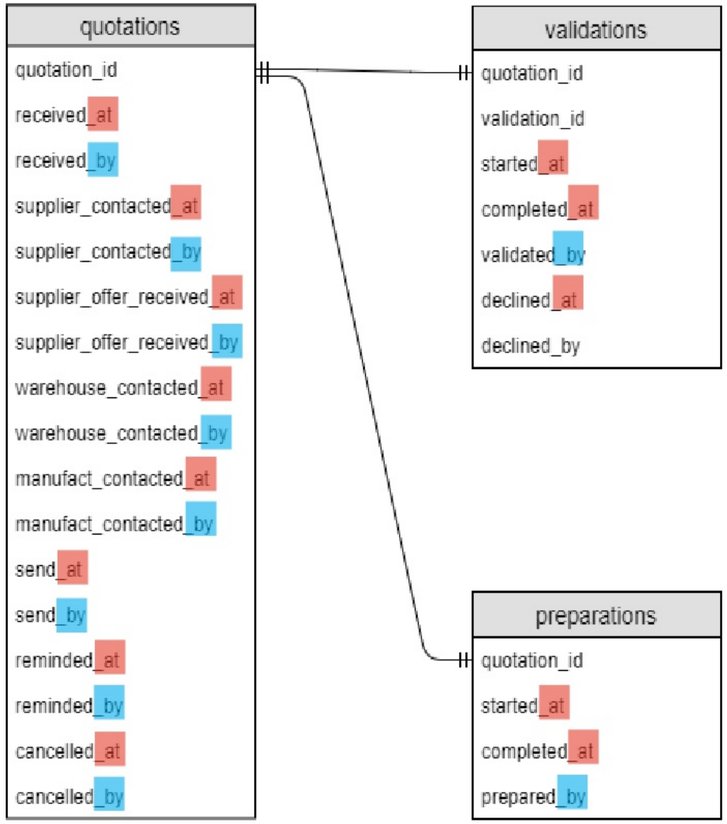

quotations
quotation_id
received_at
received_by
supplier_contacted_at
supplier_contacted_by
supplier_offer_received_at
supplier_offer_received_by
warehouse_contacted_at
warehouse_contacted_by
manufact_contacted_at
manufact_contacted_by
send_at
send_by
reminded_at
reminded_by
cancelled_at
cancelled_by


quotations
quotation_id
received_at
received_by
supplier_contacted_at
supplier_contacted_by
supplier_offer_received_at
supplier_offer_received_by
warehouse_contacted_at
warehouse_contacted_by
manufact_contacted_at
manufact_contacted_by
send_at
send_by
reminded_at
reminded_by
cancelled_at
cancelled_by
validations
quotation_id
validation_id
started_at
completed_at
validated_by
declined_at
declined_by


quotations
quotation_id
received_at
received_by
supplier_contacted_at
supplier_contacted_by
supplier_offer_received_at
supplier_offer_received_by
warehouse_contacted_at
warehouse_contacted_by
manufact_contacted_at
manufact_contacted_by
send_at
send_by
reminded_at
reminded_by
cancelled_at
cancelled_by
validations
quotation_id
validation_id
started_at
completed_at
validated_by
declined_at
declined_by
preparations
quotation_id
started_at
completed_at
prepared_by

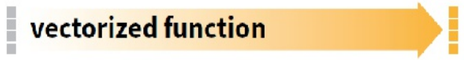

quotations
quotation_id
received_at
received_by
supplier_contacted_at
supplier_contacted_by
supplier_offer_received_at
supplier_offer_received_by
warehouse_contacted_at
warehouse_contacted_by
manufact_contacted_at
manufact_contacted_by
send_at
send_by
reminded_at
reminded_by
cancelled_at
cancelled_by
validations
quotation_id
validation_id
started_at
completed_at
validated_by
declined_at
declined_by
preparations
quotation_id
started_at
completed_at
prepared_by

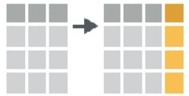

quotations
quotation_id
received_at
received_by
supplier_contacted_at
supplier_contacted_by
supplier_offer_received_at
supplier_offer_received_by
warehouse_contacted_at
warehouse_contacted_by
manufact_contacted_at
manufact_contacted_by
send_at
send_by
reminded_at
reminded_by
cancelled_at
cancelled_by
validations
quotation_id
validation_id
started_at
completed_at
validated_by
declined_at
declined_by
preparations
quotation_id
started_at
completed_at
prepared_by
Receive Request For Quotation
Contact Supplier
Receive Supplier Offer
Contact Manufacturing
Contact Warehouse
Send Quotation
Reminded Pending Notation
Cancel Quotation
Validate Request
Decline Request
Prepare Quotation


quotations
quotation_id
received_at
received_by
supplier_contacted_at
supplier_contacted_by
supplier_offer_received_at
supplier_offer_received_by
warehouse_contacted_at
warehouse_contacted_by
manufact_contacted_at
manufact_contacted_by
send_at
send_by
reminded_at
reminded_by
cancelled_at
cancelled_by
validations
quotation_id
validation_id
started_at
completed_at
validated_by
declined_at
declined_by
preparations
quotation_id
started_at
completed_at
prepared_by

MAKE NEW VARIABLES

These apply **vectorized functions** to columns. Vectorized funs take vectors as input and return vectors of the same length as output (see back).





mutate(.data, …**)**
Compute new column(s).
mutate(mtcars, gpm = 1/mpg)

COMBINE CASES

bind_rows(…, .id = NULL**)**
Returns tables one on top of the other as a single table. Set .id to a column name to add a column of the original table names (as pictured)

COMBINE VARIABLES

Use a "**Mutating Join**" to join one table to columns from another, matching values with the rows that they correspond to. Each join retains a different combination of values from the tables.

left_join(x, y, by = NULL, copy=FALSE, suffix=c(".x",".y"),…**)**
Join matching values from y to x.

right_join(x, y, by = NULL, copy = FALSE, suffix=c(".x",".y"),…**)**
Join matching values from x to y.

inner_join(x, y, by = NULL, copy = FALSE, suffix=c(".x",".y"),…**)**
Join data. Retain only rows with matches.

full_join(x, y, by = NULL, copy=FALSE, suffix=c(".x",".y"),…**)**
Join data. Retain all values, all rows.

Reshape Data - change the layout of values in a table

Use **gather()** and **spread()** to reorganize the values of a table into a new layout.

gather(data, key, value, ..., na.rm = FALSE, convert = FALSE, factor_key = FALSE**)**

gather() moves column names into a **key** column, gathering the column values into a single **value** column.

table4a

country	1999	2000
A	0.7K	2K
B	37K	80K
C	212K	213K

→

country	year	cases
A	1999	0.7K
B	1999	37K
C	1999	212K
A	2000	2K
B	2000	80K
C	2000	213K

key value

gather(table4a, `1999`, `2000`, key = "year", value = "cases")

spread(data, key, value, fill = NA, convert = FALSE, drop = TRUE, sep = NULL**)**

spread() moves the unique values of a **key** column into the column names, spreading the values of a **value** column across the new columns.

table2

country	year	type	count
A	1999	cases	0.7K
A	1999	pop	19M
A	2000	cases	2K
A	2000	pop	20M
B	1999	cases	37K
B	1999	pop	172M
B	2000	cases	80K
B	2000	pop	174M
C	1999	cases	212K
C	1999	pop	1T
C	2000	cases	213K
C	2000	pop	1T

key value

→

country	year	cases	pop
A	1999	0.7K	19M
A	2000	2K	20M
B	1999	37K	172M
B	2000	80K	174M
C	1999	212K	1T
C	2000	213K	1T

spread(table2, type, count)

BUSINESS PROCESS ANALYTICS IN R

Let's practice!

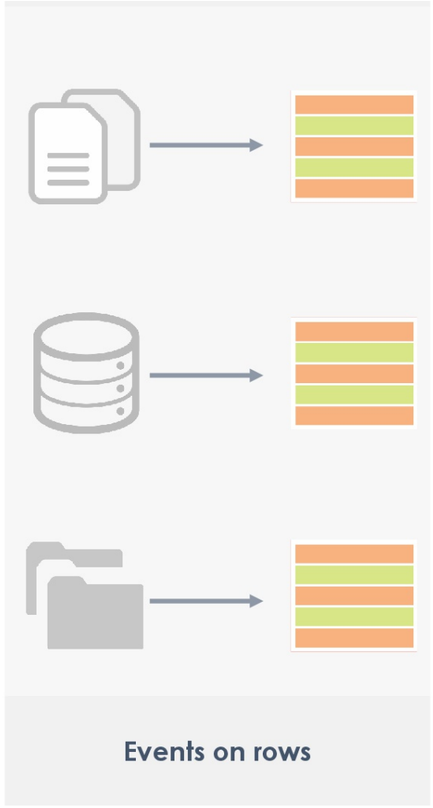
BUSINESS PROCESS ANALYTICS IN R

Getting to know the process

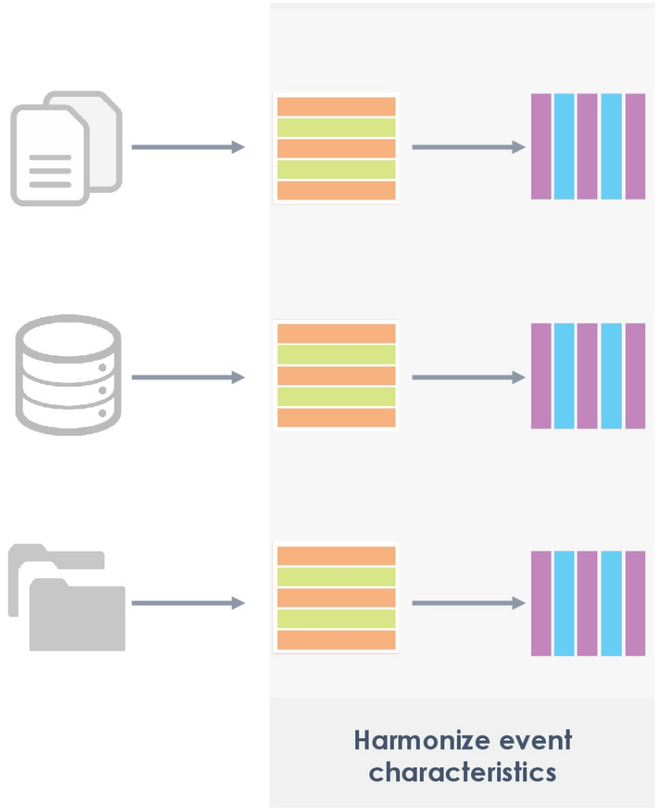
Gert Janssenswillen
Creator of bupaR

Identify data sources

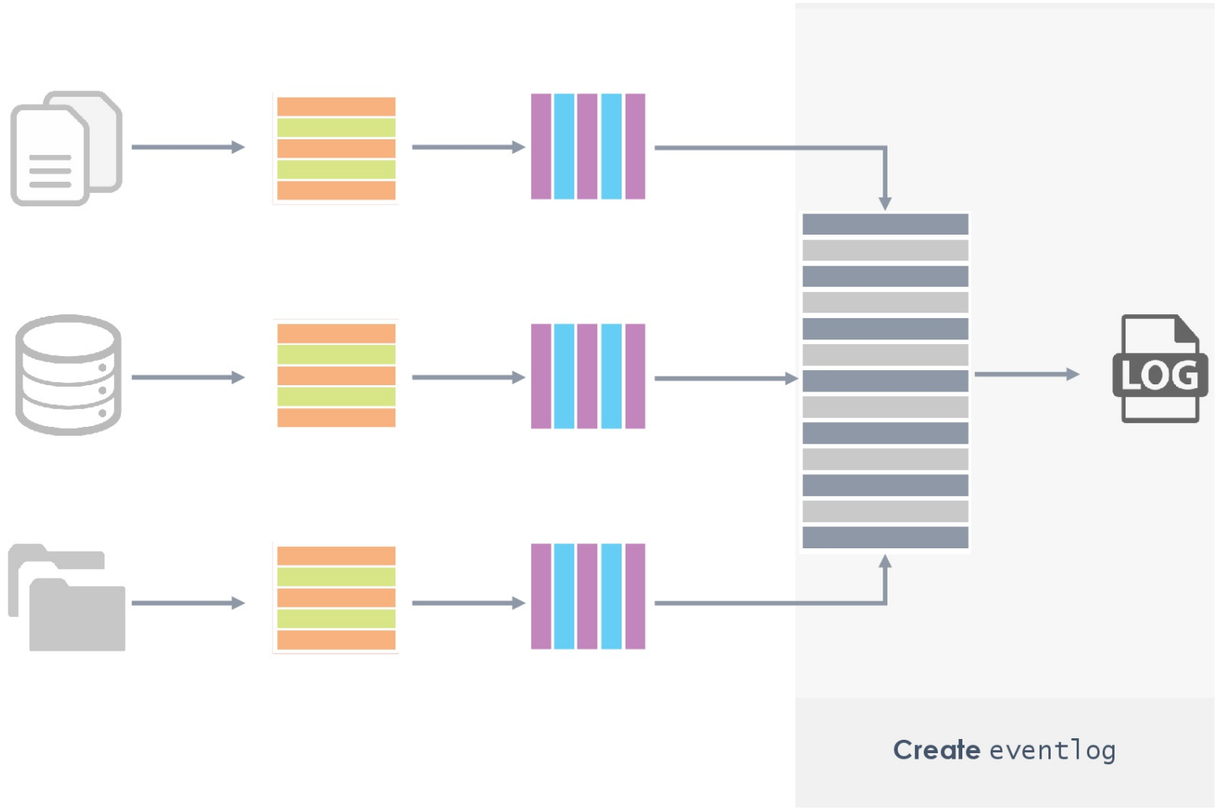
Reshape them



Harmonize them



Create event data



Inspect event data

```
summary(otc)
```

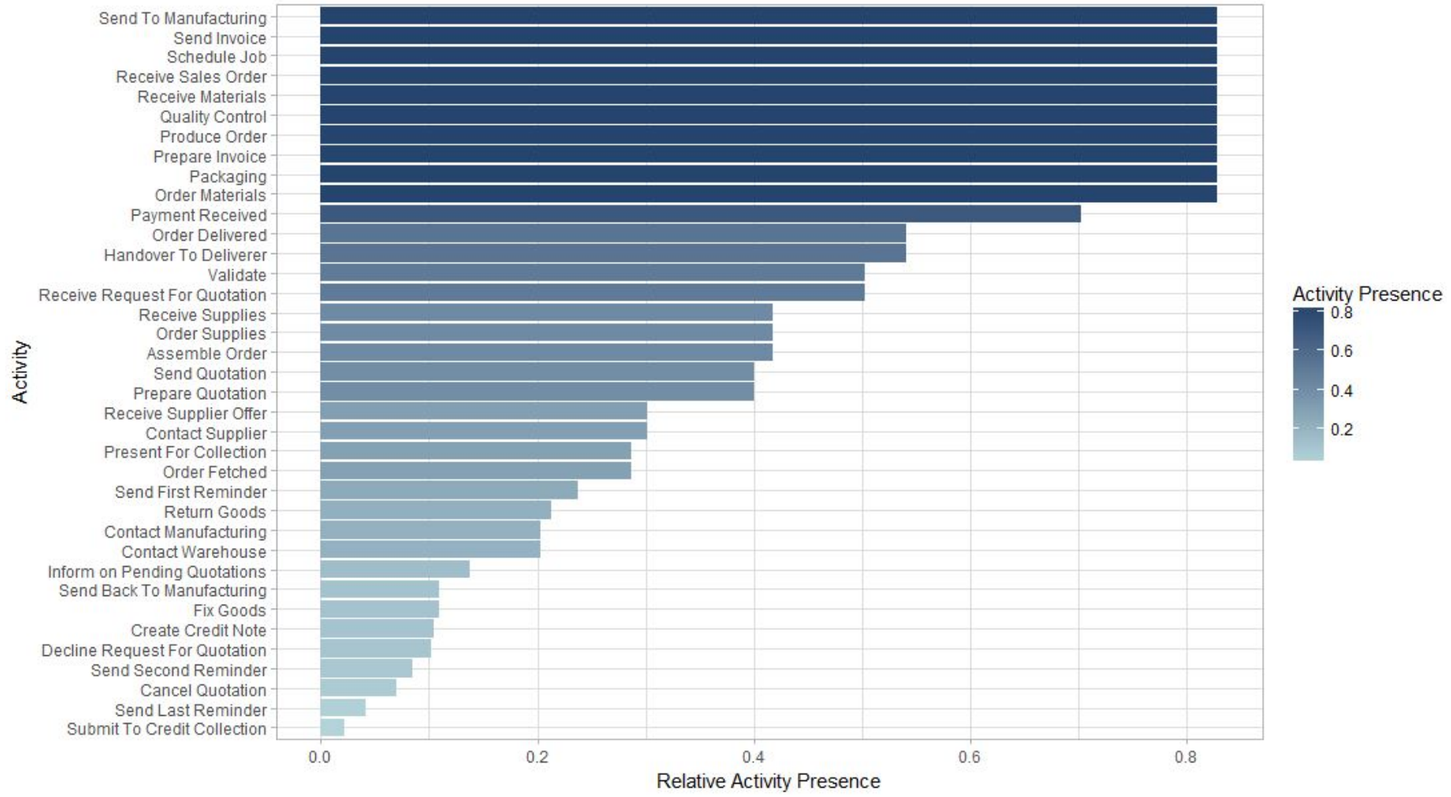
```
Number of events:  75499
Number of cases:  3651
Number of traces:  1656
Number of distinct activities:  37
Average trace length:  20.67899

Start eventlog:  2017-01-01 12:13:56
End eventlog:  2018-04-22 19:56:13

 quotation_id          resource                activity        action             lifecycle
 Length:75499       George  : 7186   Packaging       : 6048   Length:75499       complete:25420
 Class :character   Andrea  : 6594   Prepare Invoice: 6048   Class :character   start   :50079
 Mode  :character   Virginia: 6235   Produce Order   : 6048   Mode  :character
                    Jonathan: 6047   Quality Control: 6048
                    Cheryl  : 5381   Validate        : 3666
                    Louis   : 5039   Assemble Order  : 3048
                    (Other) :39017   (Other)         :44593
   timestamp                    sales_order_id      case_id              .order
 Min.   :2017-01-01 12:13:56   Length:75499       Length:75499       Min.   :    1
 1st Qu.:2017-05-07 14:26:12   Class :character   Class :character   1st Qu.:18876
 Median :2017-08-07 15:30:29   Mode  :character   Mode  :character   Median :37750
 Mean   :2017-08-07 23:16:03                                         Mean   :37750
 3rd Qu.:2017-11-09 17:27:47                                         3rd Qu.:56625
 Max.   :2018-04-22 19:56:13                                         Max.   :75499
```

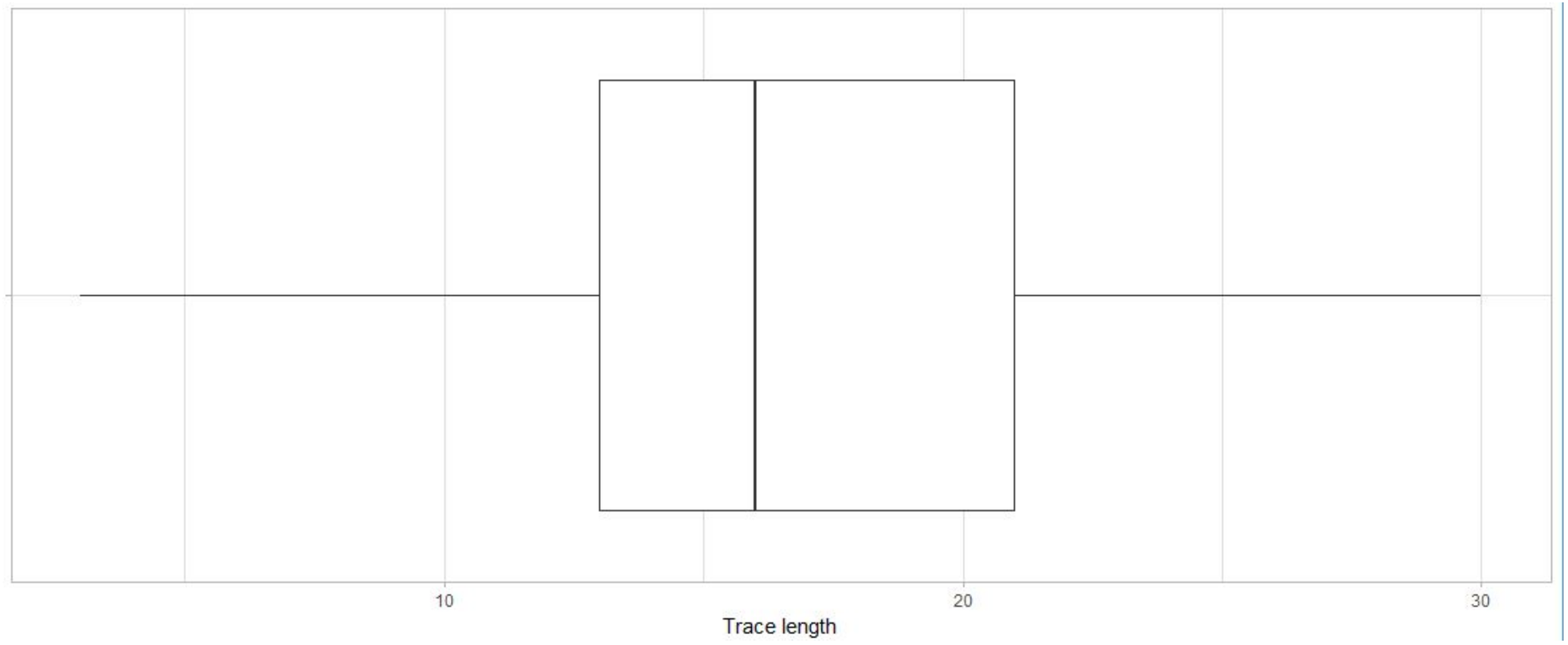
Activity presence

```
activity_presence(otc) %>% plot()
```



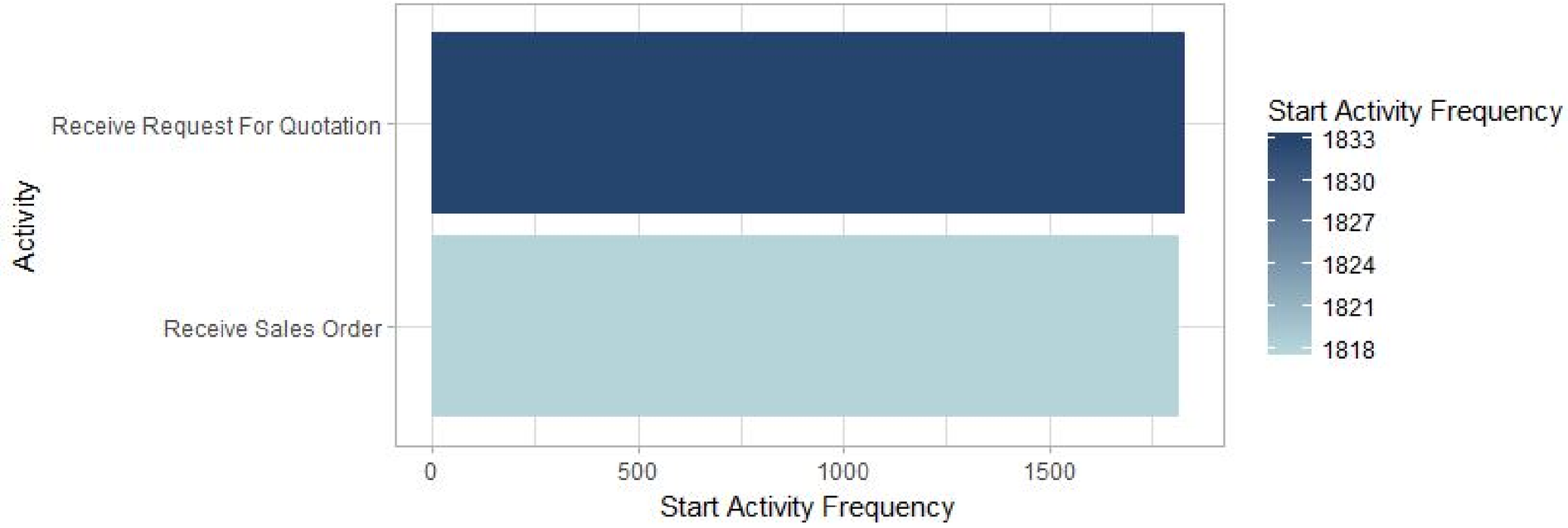
Trace lengths

```
trace_length(otc) %>% plot()
```



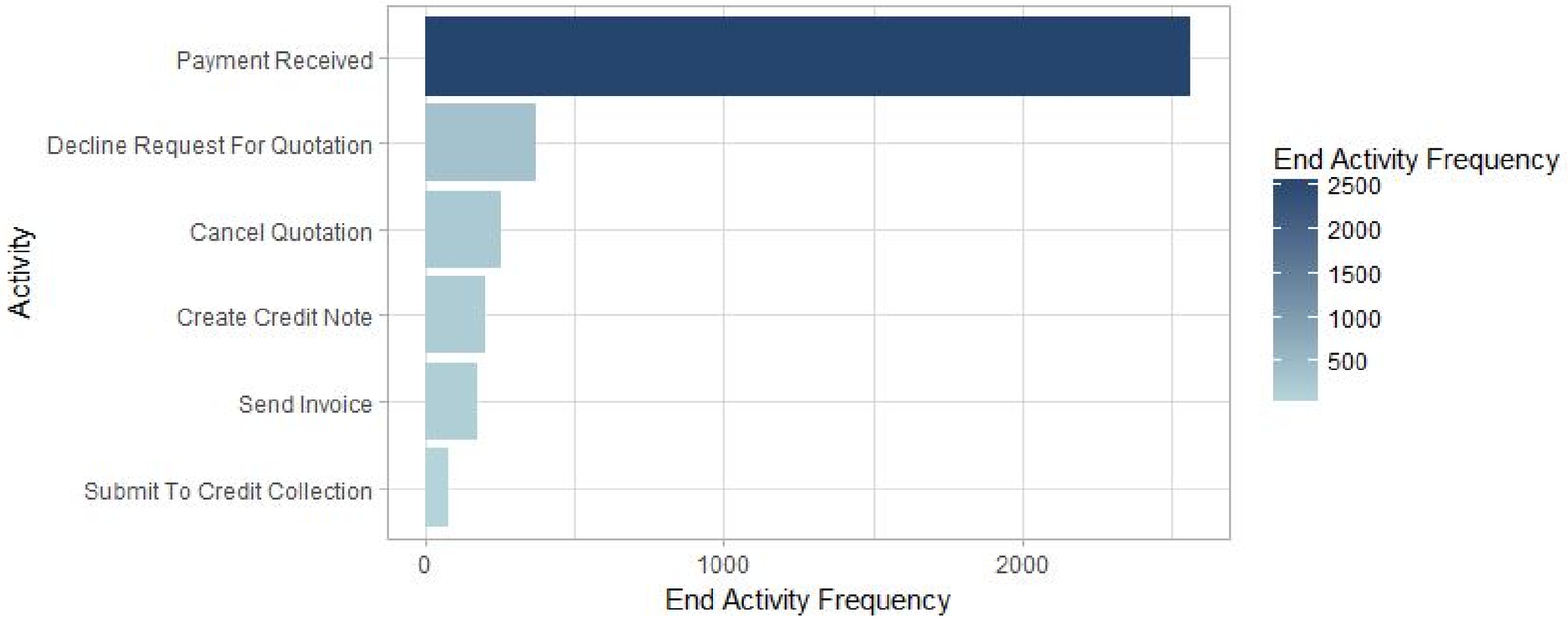
Entry points

```
start_activities(otc, "activity") %>% plot()
```



Exit points

```
end_activities(otc, "activity") %>% plot()
```



BUSINESS PROCESS ANALYTICS IN R

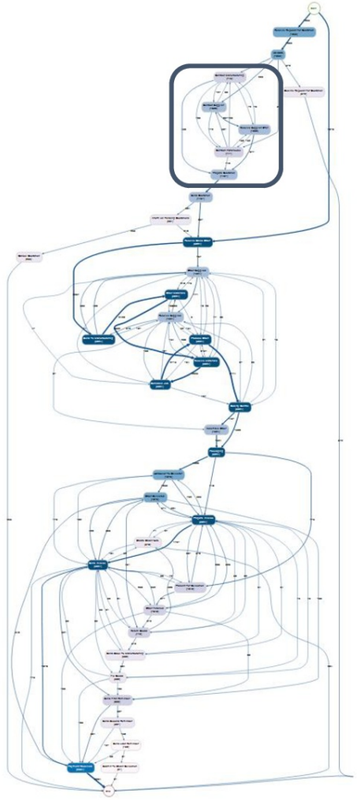
Let's practice!

BUSINESS PROCESS ANALYTICS IN R

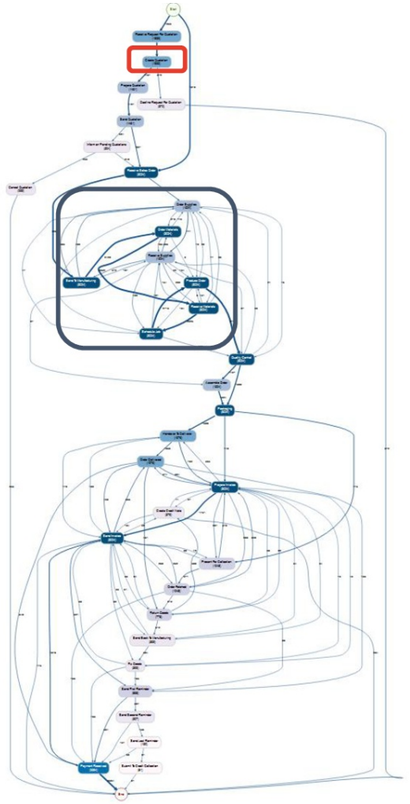
Roles & Rules

Gert Janssenswillen
Instructor

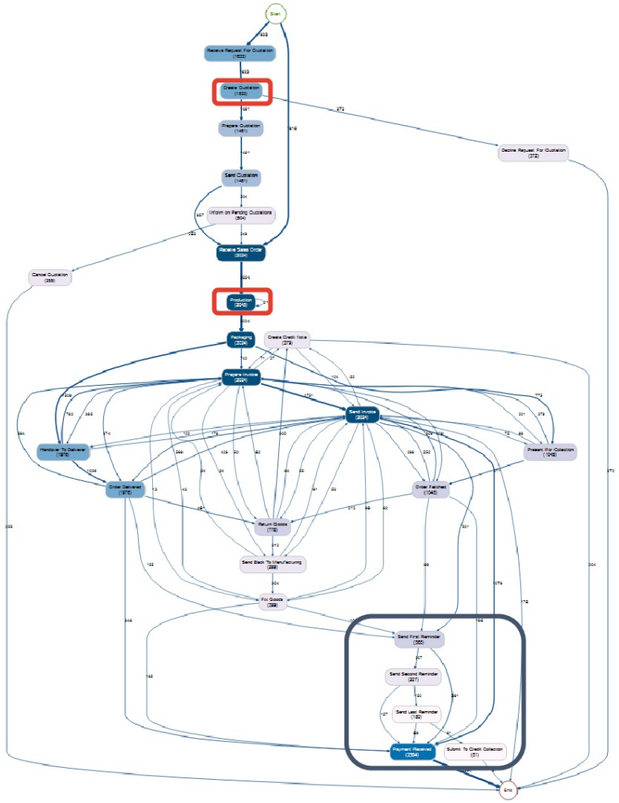
Zooming out



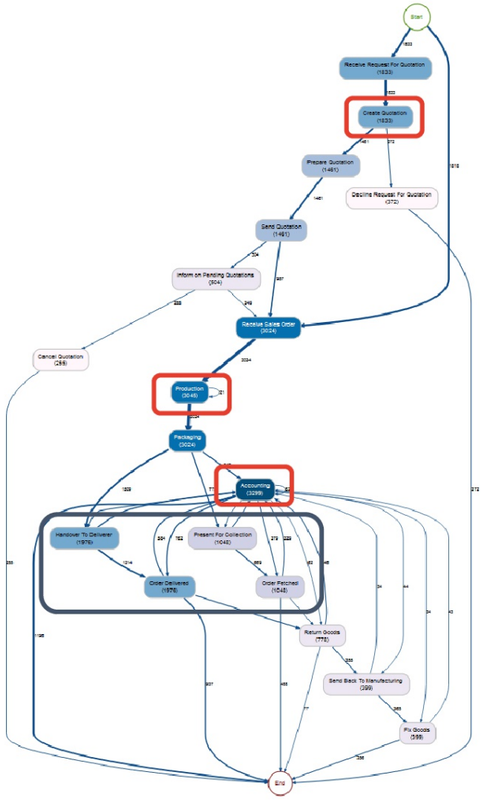
Zooming out



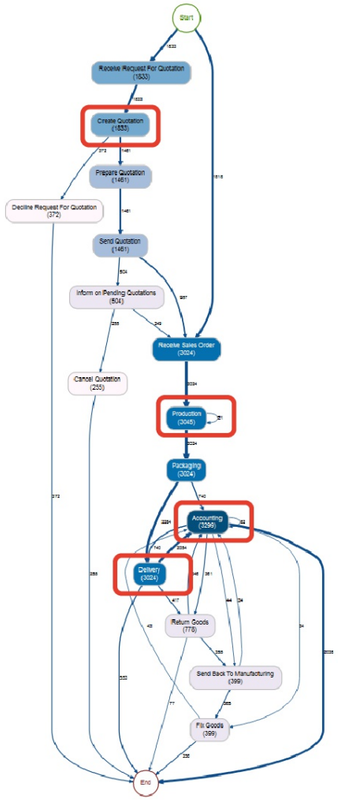
Zooming out


Start
Receive Request For Quotation
Create Quotation
Decline Request For Quotation
Prepare Quotation
Send Quotation
Inform on Pending Quotations
Delete Quotation
Receive Sales Order
Reactivate
Packaging
Create Credit Note
Prepare Invoice
Send Invoice
Reminder To Declare
Present For Collection
Order Declared
Order Fetched
Return Goods
Send Back To Manufacturing
Fix Goods
Send First Reminder
Send Second Reminder
Send Last Reminder
Payment Received
Submit To Credit Collection
End

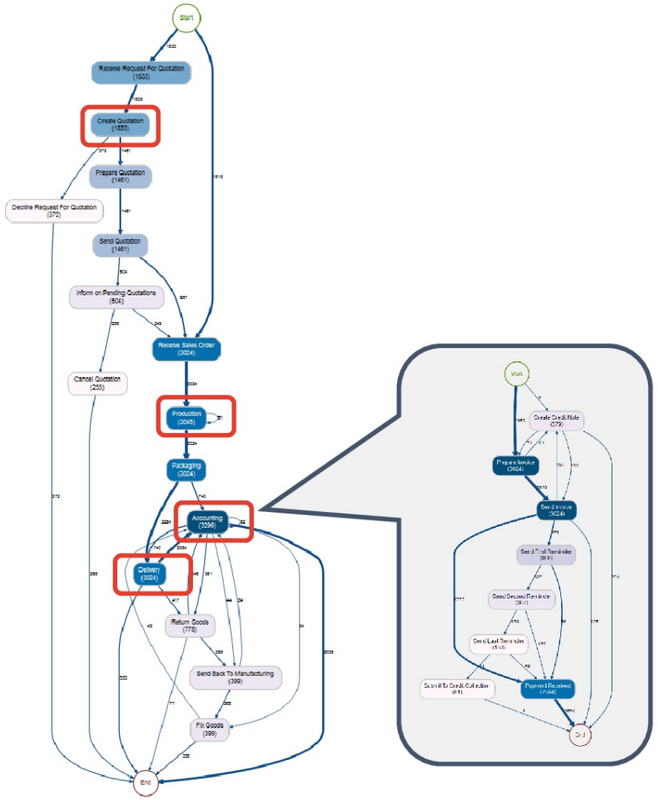
Zooming in


Start
Receive Request For Quotation (1633)
Create Quotation (1633)
Prepare Quotation (1461)
Decline Request For Quotation (372)
Send Quotation (1461)
Inform on Pending Quotations (924)
Receive Sales Order (3024)
Cancel Quotation (249)
Production (3048)
Packaging (3024)
Accounting (3299)
Handover To Deliverer (1976)
Present For Collection (1042)
Order Delivered (1976)
Order Fetched (1042)
Return Goods (775)
Send Back To Manufacturing (399)
Fix Goods (595)
End

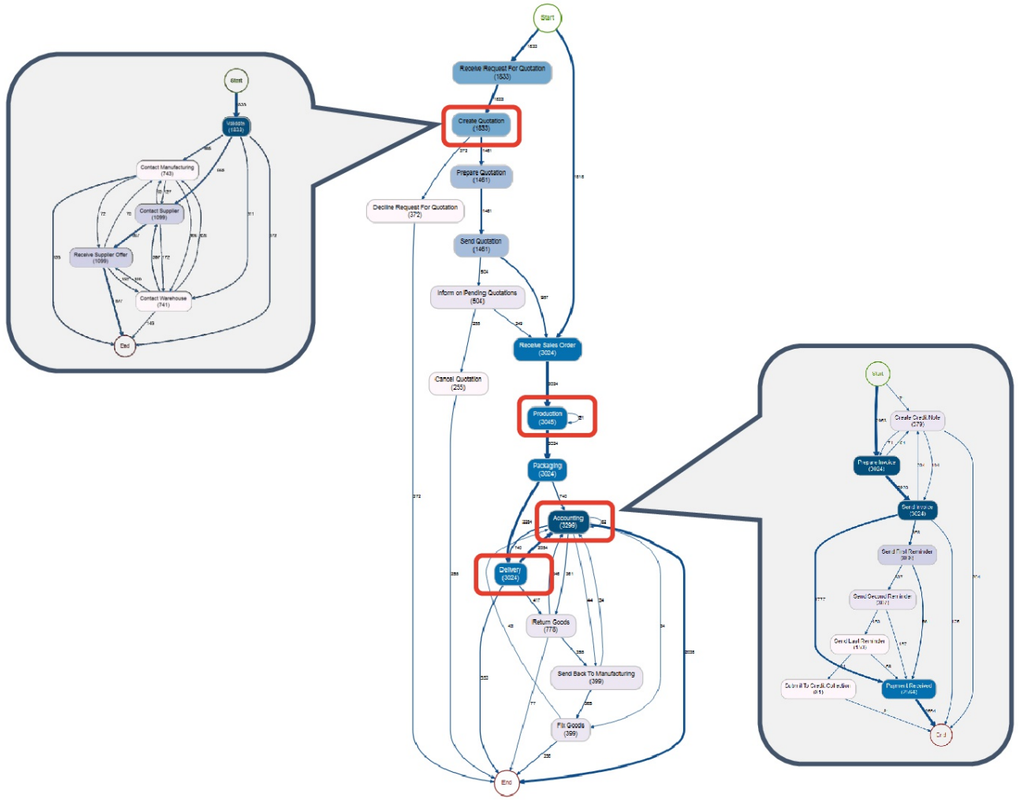
Zooming out


Start
Release Request For Quotation (1633)
Create Quotation (1633)
Prepare Quotation (1461)
Decline Request For Quotation (372)
Send Quotation (1461)
Inform on Pending Quotations (504)
Receive Sales Order (3024)
Cancel Quotation (255)
Production (3048)
Packaging (3024)
Accounting (3056)
Deliver (3024)
Return Goods (778)
Send Back To Manufacturing (399)
Fix Goods (399)
End

Zooming in


Start
Release Request For Quotation (1633)
Create Quotation (1633)
Prepare Quotation (1481)
Decline Request For Quotation (372)
Send Quotation (1481)
Inform on Pending Quotations (504)
Receive Sales Order (3024)
Cancel Quotation (255)
Production (3048)
Packaging (3024)
Accounting (3556)
Deliver (3024)
Return Goods (778)
Send Back To Manufacturing (399)
Fix Goods (399)
End

Zooming in


Start
Release Request For Quotation (1633)
Create Quotation (1633)
Prepare Quotation (1481)
Decline Request For Quotation (372)
Send Quotation (1481)
Inform on Pending Quotations (504)
Receive Sales Order (3024)
Cancel Quotation (255)
Production (3048)
Packaging (3024)
Accounting (3056)
Deliver (3024)
Return Goods (778)
Send Back To Manufacturing (399)
Fix Goods (399)
End
Contact Manufacturing (743)
Contact Supplier (1099)
Receive Supplier Offer (1099)
Contact Warehouse (741)

Answering Research Questions

Performance

Answering Research Questions

Performance

Compliance

Roles & Rules

- Resource Roles
- 4-eye principle
- Control-flow deviations

BUSINESS PROCESS ANALYTICS IN R

Let's practice!

BUSINESS PROCESS ANALYTICS IN R

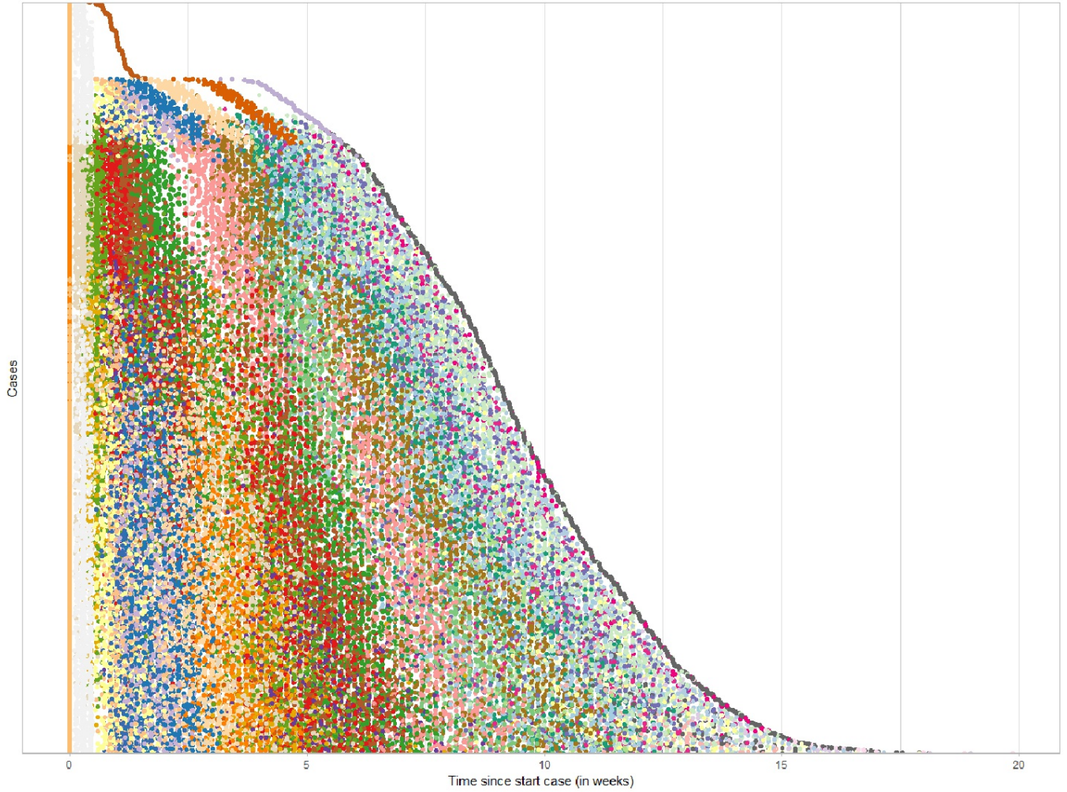
Fast Production, Fast Delivery

Gert Janssenswillen
Creator of bupaR

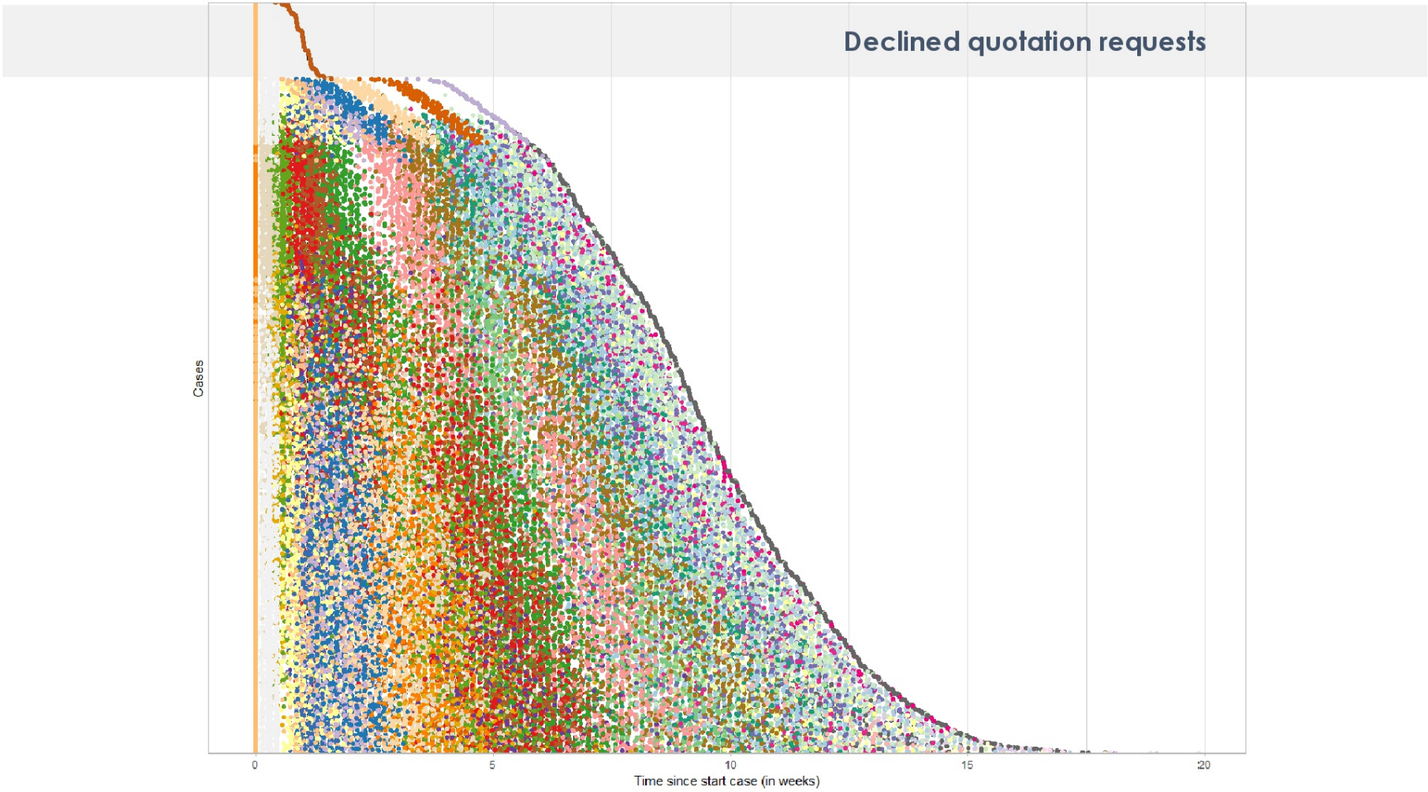
Dotted chart

```
otc %>% dotted_chart(x = "relative", sort = "duration")
```

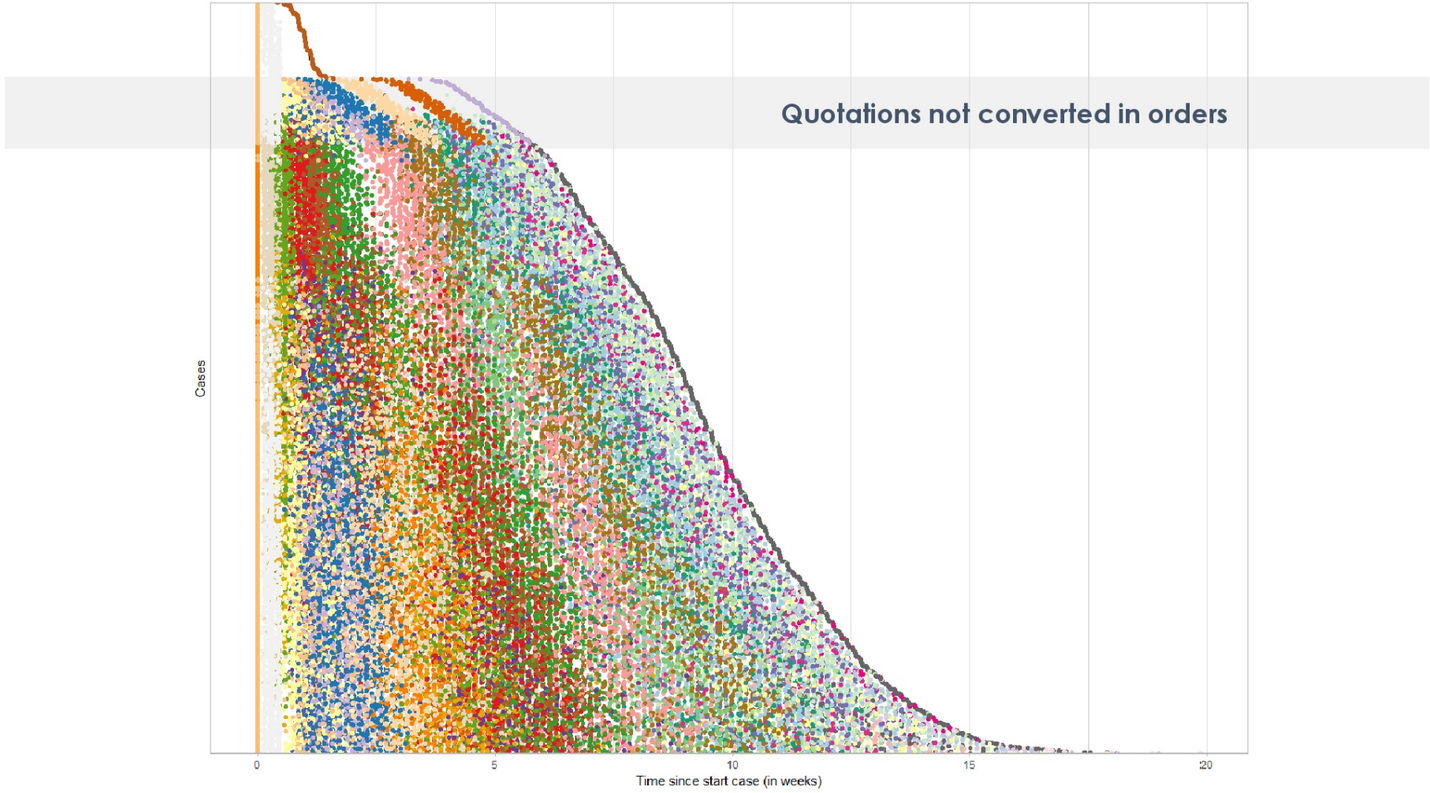
Dotted chart



Dotted chart

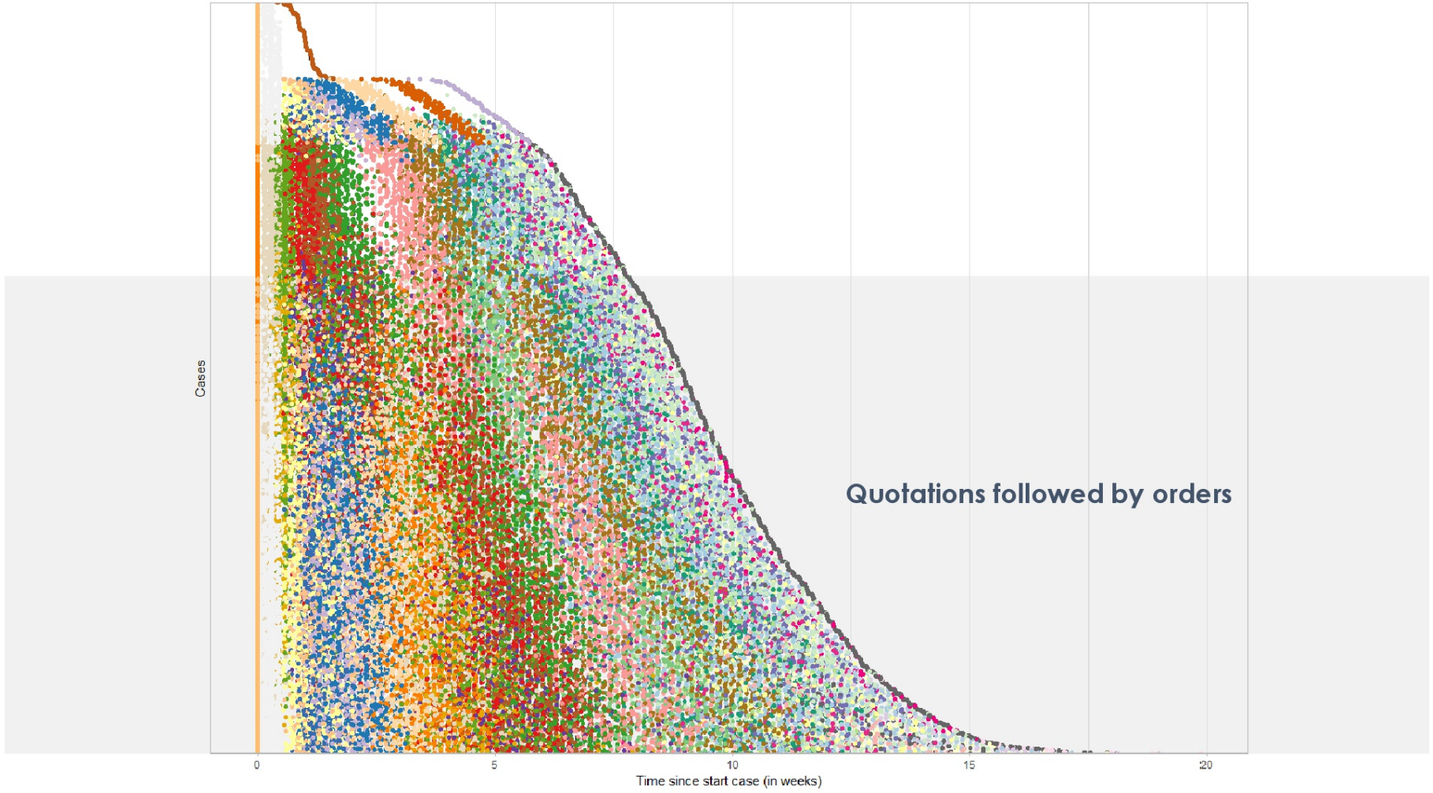


Dotted chart

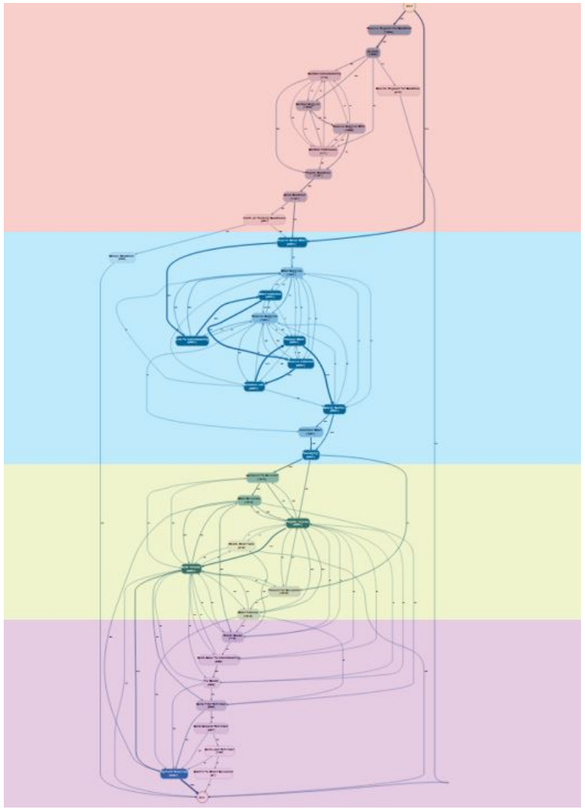


Dotted chart

Dotted chart



Process stages



BUSINESS PROCESS ANALYTICS IN R

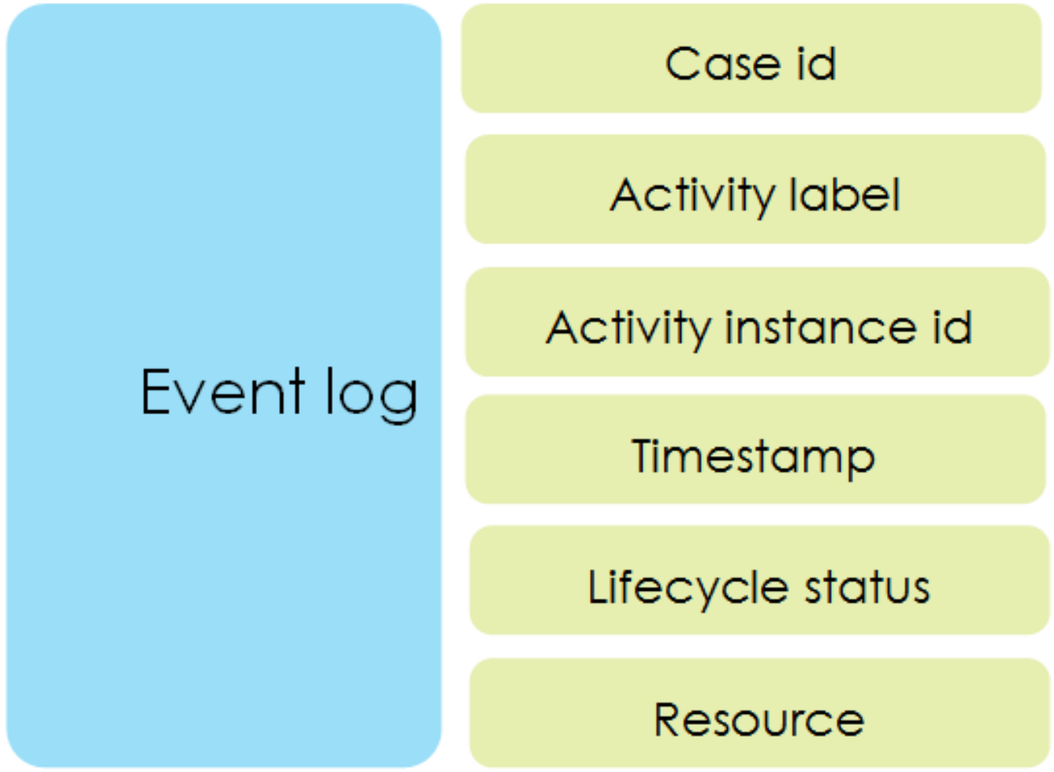
Let's practice!

BUSINESS PROCESS ANALYTICS IN R

Congratulations

Gert Janssenswillen
Creator of bupaR

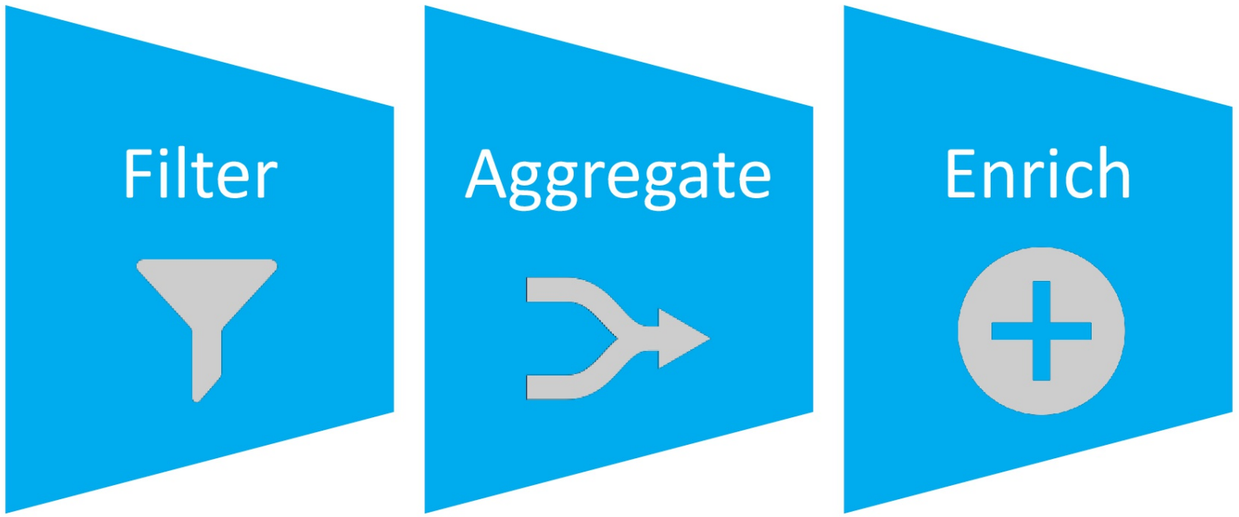
Process data



Process Analytics

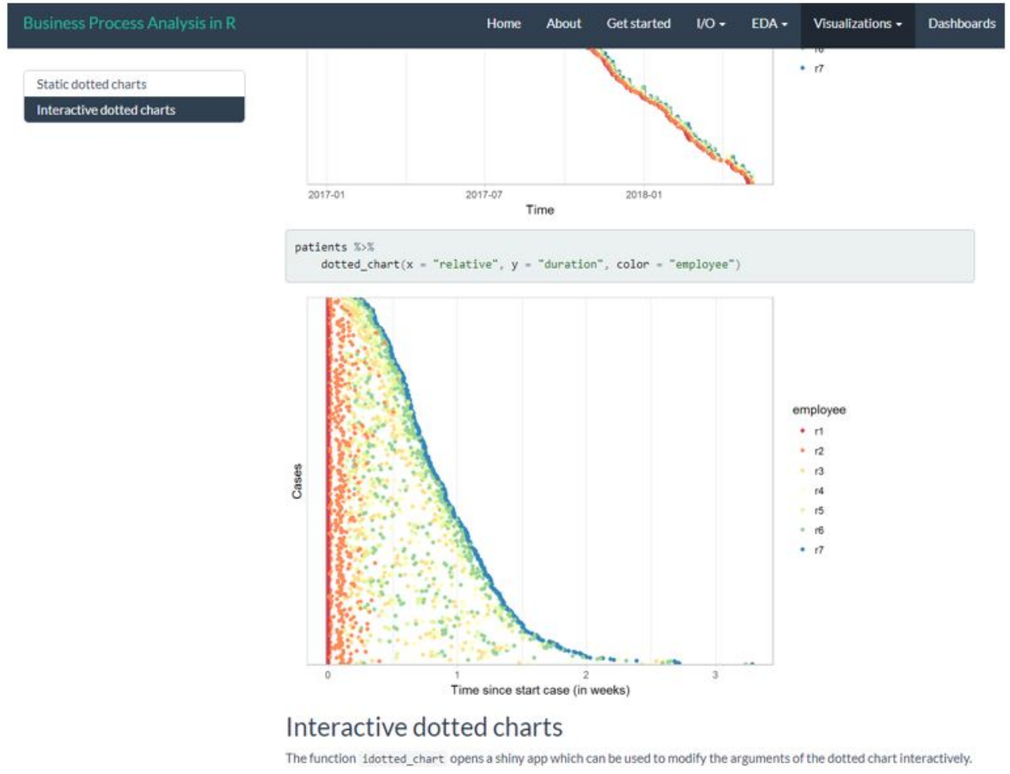
	Organizational	Structuredness	Performance
Metrics	Resource Frequency Resource Specialization Resources Involvement	Start Activities End Activities Trace Coverage Trace Length Repetitions Self-loops Activity Presence	Processing Time Throughput Time Idle Time
Visuals	Resource Map	Process Map Trace Explorer Precedence Matrix	Performance Map Dotted Chart

Process Data Preprocessing



There's more

www.bupar.net



BUSINESS PROCESS ANALYTICS IN R

Congratulations!